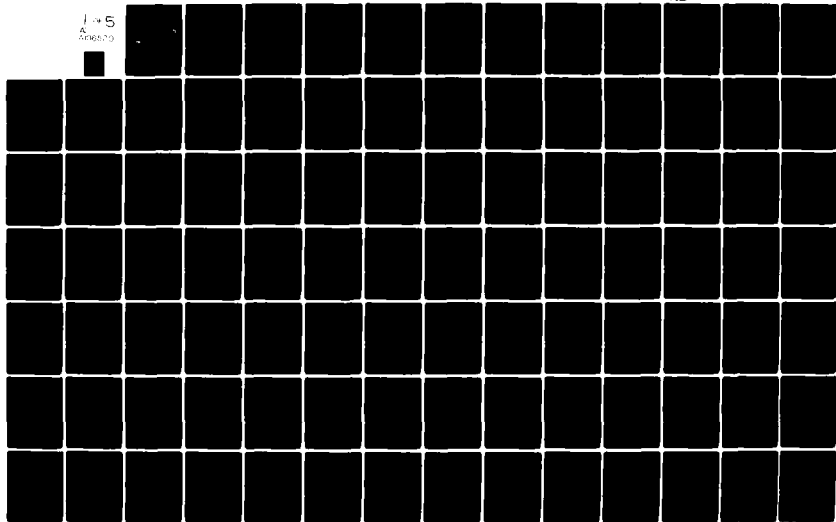


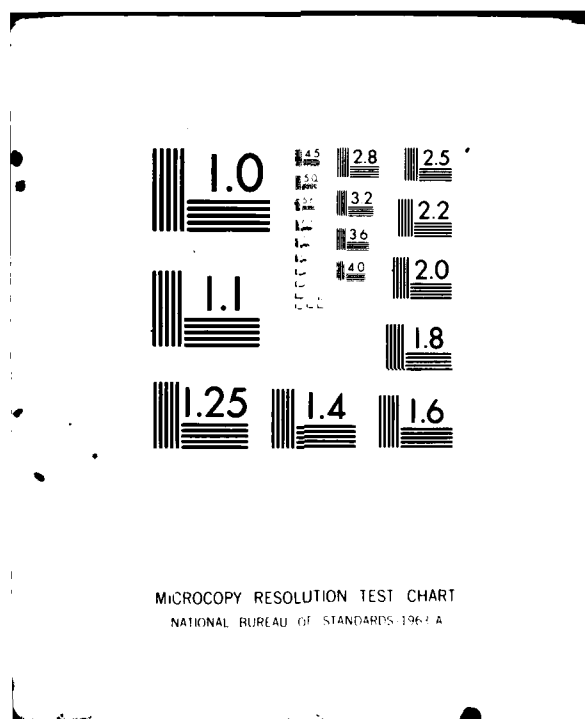
AD-A106 520

DOUGLAS AIRCRAFT CO. LONG BEACH CA
NUCLEAR BLAST RESPONSE COMPUTER PROGRAM. VOLUME III. PROGRAM LI--ETC(U)
AUG 81 J A MCOREN, H H CROXEN, T P KALMAN DAA001-75-C-0216
AFWL-TR-81-32-VOL-3 ML

UNCLASSIFIED

1-5
006520





LEVEL

2

AFWL-TR-81-32, Vol. III

AFWL-TR-
81-32
Vol. III

AD A106520

NUCLEAR BLAST RESPONSE COMPUTER PROGRAM

Volume III of III
Program Listing

J. A. McGrew, et al.

Douglas Aircraft Company
3855 Lakewood Blvd.
Long Beach, CA 90846

August 1981

Final Report

Approved for public release; distribution unlimited.

THIS RESEARCH WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY UNDER
SUBTASK N99AXAE500, WORK UNIT 04, WORK UNIT TITLE: NUCLEAR BLAST
RESPONSE COMPUTER PROGRAM.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

DTIC
ELECTE
NOV 2 1981
H

DTIC FILE COPY

81 11 02 064

This final report was prepared by the Douglas Aircraft Company, Long Beach, California under Contracts DNA 001-75-C-0216 and DNA 001-76-C-0346, Job Order 88090340 with the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. Mr. Alfred L. Sharp (NTYV) was the Laboratory Project Officer-in-Charge.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by a contractor of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

If your address has changed, if you wish to be removed from our mailing list, or if your organization no longer employs the addressee, please notify AFWL/NTYV, Kirtland AFB, NM 87117 to help us maintain a current mailing list.

This report has been reviewed and is approved for publication.

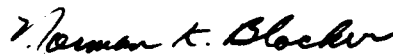


ALFRED L. SHARP
Project Officer

FOR THE COMMANDER



KENNEDY B. WILSON
Lt Colonel, USAF
Chief, Environment & Effects Branch



NORMAN K. BLOCKER
Colonel, USAF
Chief, Applied Physics Division

DO NOT RETURN COPIES OF THIS REPORT UNLESS CONTRACTUAL OBLIGATIONS OR NOTICE ON A SPECIFIC DOCUMENT REQUIRES THAT IT BE RETURNED.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL TR-81-32 VOL-3 VOL. III of III	2. GOVT ACCESSION NO. AD-A106 520	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NUCLEAR BLAST RESPONSE COMPUTER PROGRAM Volume III, Program Listing		5. TYPE OF REPORT & PERIOD COVERED Final Report, May 7, 1981
7. AUTHOR(s) J. A. McGrew, H. H. Croxen, T. P. Kalman, J. P. Giesing, W. P. Rodden		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Douglas Aircraft Company 3855 Lakewood Blvd. Long Beach, CA 90846		8. CONTRACT OR GRANT NUMBER(s) DNA 001-75-C-0216 and DNA 001-76-C-0346
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/88090340
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Director Defense Nuclear Agency Washington, DC 20305		12. REPORT DATE August 1981
		13. NUMBER OF PAGES 434
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This research was sponsored by the Defense Nuclear Agency under Subtask N99AXAE500, Work Unit 04, Work Unit Title: Nuclear Blast Response Computer Program. (over)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Aircraft Structural Response Blast Mass Modeling Nuclear Vulne. Aerodynamic Modeling Dynamic Loads Digital Computer Program Gust Response		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number.) The VIBRA-6 computer program is a digital computer program developed to determine the response of aircraft to nuclear explosions when flying at sub- sonic speeds. It is similar to the VIBRA-4 program but uses the latest Doublet-Lattice Method for obtaining subsonic aerodynamic forces for arbitrary lifting surface-body configurations. The Doublet-Lattice procedure has been extended to account for the moving blast wave by considering it as a traveling gust. The nuclear blast representation remains the same as that used in the		

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

18. SUPPLEMENTARY NOTES (Continued)

This report is divided into three volumes: Volume I contains the overall program descriptions and method of analysis, the input and output data descriptions, the program operation and a sample problem. Volume II details the unsteady aerodynamic procedure and Volume III contains the program listings. ~~★~~

20. ABSTRACT (Continued)

VIBRA-4 program but the method of solution of the equations of motion has been changed from that of numerical integration of quasi-steady equations of motion to a Fourier transform procedure to move from frequency domain solutions to time history solutions. / The concept of dynamic core has been introduced to the program thus removing any restrictions on the size of the aircraft idealization which can be analyzed.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This report was prepared by the Douglas Aircraft Company, Long Beach, California, under Contract DNA 001-75-C-0216 and documents the subsonic unsteady aerodynamic module development for the revised VIBRA-6 Nuclear Blast Response Computer Program. This work was performed under Program Element NWE D 62704H, Project N99QAXA, Task Area B500, Work Unit 04 and was funded by the Defense Nuclear Agency under: RDT & E RMSS Code B342075464N99QAXAE50004H2590D. Funding of this effort was also supported by the Air Force Weapons Laboratory under: Program Element 62601F, Project 8809, Task 03, Work Unit 40.

Inclusive dates of research and development as documented herein were May 1975 through June 1977.

Volume I of this report documents the overall program descriptions and method of analysis, the input and output data descriptions, the program operation and a sample problem. Volume II of this report details the unsteady aerodynamic procedure.

J. A. McGrew was the program technical director for this task. The technical development was performed by J. P. Giesing and T. P. Kalman with the assistance of Dr. W. P. Rodden. The programming effort was carried out by T. P. Kalman and H. H. Croxen.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Mail and/or	
Dist	Special
A	

CONTENTS

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
CONTROL		1
	DACGUST	
	ATMOS	
	FDDTBL	
	HEADNG	
	MIS1	
	MIS2	
	PLT1	
	PRNT	
	RDTAPE	
	TRAERO	
	WRAERO	
	WRFRSP	
	WRLOAD	
	WRTAPE	
	WRUNIT	
AERODYNAMIC		45
	CSDLM	
	AERO	
	ATAN3	
	BFSMAT	
	DOTP	
	DPPS	
	DPZY	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
	DUMULT	
	DYPZ	
	DZPY	
	DZYMAT	
	DZY	
	FLLD	
	FWMW	
	FZY2	
	GEND	
	GENF	
	GEOM	
	GRUP	
	GUST	
	HATS	
	IDF1	
	IDF2	
	INCRO	
	INPUTA	
	MATMUL	
	MUZYC	
	NACL	
	NEWH	
	ORGN	
	PISTON	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
	READD	
	RHSIDE	
	ROWDYZ	
	RWREC	
	SB	
	SDLM	
	SNPDF	
	SOLVIT	
	SORT	
	SPLINE	
	SPLN	
	SPL1	
	SPL2	
	SUBB	
	SUBI	
	SUBP	
	TKER	
	TVOR	
	WANDWT	
	ZEROUT	
INERTIAL		226
	GENMS	
	INERTM	
	INPUTM	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
	INPUTS	
	LHS	
	TMST	
	TMSTL	
UNIT LOADS		239
	LOAD	
	ABAMLT	
	AROLOD	
	AROPHA	
	BEAM	
	GEOMAB	
	GEOMMB	
	GEOMSB	
	INPUTL	
	MMBMLT	
	MSSPHI	
	RTOI	
	SBAMLT	
	SYMSOL	
	THRUST	
ACTIVE CONTROL		274
	ACS	
	ACSMOD	
	AFCS	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
	TFBLIN	
	TFBLOK	
	TRFVAL	
FREQUENCY RESPONSE		286
	CFREQR	
	COEFF	
	FORMC	
	FREQRS	
	GSTINI	
	INTERP	
	TDINT	
	TFMATF	
	TFMATM	
	TRFCMX	
UNIT GUST LOADS		309
	CFRLOD	
	FGMULT	
	FRLOAD	
	INTLPR	
TRIM		324
	AERO2	
	CTRIM	
	SPANLD	
	TRIM	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
BLAST AND TIME RESPONSE		347
	ACCHST	
	CGUST	
	FLTPOS	
	GSTHST	
	GUSRHO	
	GUSTDR	
	IFT	
	INTRPB	
	LOADCH	
	PRESS	
	RANGE	
	STRSCH	
	TAR	
	TIMHST	
	TPEVAL	
	TPINT	
	TRFFT	
RIGID		396
	BLINT	
	CRIGID	
	GSINT	
	INTRP2	
	RIGID	

CONTENTS (Cont'd)

<u>Module</u>	<u>Subroutine</u>	<u>Page</u>
MERGE		412
	CMERGE	
	COPY	
	FCOPY	
	MERGE	

```

PROGRAM DACGUST (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE31
2 , TAPE10=512, TAPE21=512, TAPE3=512, TAPE4=512, TAPE8=512, TAPE9=512
3 , TAPE11=512, TAPE12=512, TAPE13=512, TAPE14=512
4 , TAPE15=512, TAPE16=512, TAPE17=512, TAPE18=512, TAPE19=512
5 , TAPE20=512, TAPE21=512, TAPE22=512, TAPE23=512
6 , TAPE34=512, TAPE35=512, TAPE36=512 )

COMMON NAA, A(1)
COMMON/DISK2/ND2, ITBL2(843), NRECSA, IBUMP, NKD, VOBWS(20)
COMMON /ZZZ/HEDR(33), TITLE(12), DT(6), NIN, NOUT, KROW, LINES, IPRNT, NERD
COMMON /XTE/NF(100)
DIMENSION XF(100)
EQUIVALENCE (NF(1), XF(1)), XF(11),
EQUIVALENCE (XF(5), NMS),
, (XF(6), NSYM), (XF(7), NASYM), (XF(8), NOMOD),
, (XF(9), NREQ), (XF(10), NENG), (XF(11), IDENT),
, (XF(12), BZERO), (XF(13), RHO), (XF(14), SIGMA),
, (XF(15), EQUVAS), (XF(16), IUNIT), (XF(17), VKEAS),
, (XF(18), VTFPS), (XF(19), ALT),
, (XF(24), KPRLDS), (XF(25), KPRIMH)
, (XF(29), NACC),
, (XF(70), MXORD), (XF(71), MXORSN), (XF(72), MXORSN),
, (NF(73), MXGRAN), (NF(74), MXORAD), (NF(75), NTF), (NF(76), NTF),
, (XF(122), KPRBLS), (XF(20), KPRCHK), (XF(21), KPRCXI)
, (XF(100), SIZECT),
, (XF(23), KPRTRM), (XF(51), AN), (XF(52), ZDOT),
, (XF(53), RTURN), (XF(54), KMAN), (XF(55), AB),
, (XF(56), AC), (XF(57), INDSYM), (XF(61), NORMAX),
, (XF(62), TIMEV), (XF(63), EFR), (XF(64), KGRD),
, (XF(65), KLPT), (XF(66), HGRD), (XF(67), KLOAD),
, (XF(68), NCRTS),
, (XF(31), NBEAMS), (XF(32), NINTLD), (XF(33), NSTRSS),
, (XF(34), NMGRP), (XF(35), NABGRP), (XF(36), NSBGRP)
EQUIVALENCE (XF(25), IPLQ),
, (XF(27), IPLL),
, (XF(28), IPLBL),
, (XF(37), NBOXES), (XF(38), NAERSB),
EQUIVALENCE (XF(41), NK), (XF(42), NG),
, (XF(43), NBOX), (XF(44), NSBETO), (XF(45), NB),
, (XF(48), IDIMUL),
, (XF(1), AMACH),
, (XF(3), VSS),
, (XF(97), DELT),
, (XF(99), RBRADF),
, (XF(4), PO),
, (XF(96), IPRNTM)
EQUIVALENCE (XF(80), ISECT), (XF(81), NDOF), (XF(82), IMS)
COMMON /AEROMX/INTRD, NVBWMX, VOBWIN(400), RINTP(50,3)
COMMON /DOTBL/DOTBL(20,10)
INTEGER DOTBL
DIMENSION CR(3,20)
DIMENSION UNITS(3)

```

```

DACGUST12
DACGUST13
DACGUST14
DACGUST15
DACGUST16
DACGUST17
DACGUST18
DACGUST19
DISK2 2
DACGUST12
DACGUST13
DACGUST14
DACGUST15
DACGUST16
DACGUST17
DACGUST18
DACGUST19
DACGUST20
DACGUST21
DACGUST22
DACGUST23
DACGUST24
DACGUST25
DACGUST26
DACGUST27
DACGUST28
DACGUST29
DACGUST30
DACGUST31
DACGUST32
DACGUST33
DACGUST34
DACGUST35
DACGUST36
DACGUST37
DACGUST38
DACGUST39
DACGUST40
DACGUST41
DACGUST42
DACGUST43
DACGUST44
DACGUST45
DACGUST46
DACGUST47
DACGUST48
DACGUST49
DACGUST50
DACGUST51
DACGUST52
DACGUST53

```

DACGUS54
 DACGUS55
 DACGUS56
 DACGUS57
 DACGUS58
 DACGUS59
 DACGUS60
 DACGUS61
 DACGUS62
 DACGUS63
 DACGUS64
 DACGUS65
 DACGUS66
 DACGUS67
 DACGUS68
 DACGUS69
 DACGUS70
 DACGUS71
 DACGUS72
 DACGUS73
 DACGUS74
 DACGUS75
 DACGUS76
 DACGUS77
 DACGUS78
 DACGUS79
 DACGUS80
 DACGUS81
 DACGUS82
 DACGUS83
 DACGUS84
 DACGUS85
 DACGUS86
 DACGUS87
 DACGUS88
 DACGUS89
 DACGUS90
 DACGUS91
 DACGUS92
 DACGUS93
 DACGUS94
 DACGUS95
 DACGUS96
 DACGUS97
 DACGUS98
 DACGUS99
 DACGUS100
 DACGUS101
 DACGUS102
 DACGUS103
 DACGUS104
 DACGUS105

```

DIMENSION IHD(50),RHD(50)
DIMENSION OPDATA(6)
DIMENSION OPDATA(10), OPT(10)
EQUIVANCE (IHD(1),RHD(1))
DATA AERO/4HAERO/, AMOD/4HIMOD/, ALQD/4HLOAD/, SECT/4HSECT/
DATA UNITS/1.0,0.59208578,0.68181818/
DATA CORERA/8HAERO STG/
DATA BLANK/4H /
DATA OPT/4HAERO,4HUNIT,4HACSM,4HFPS,4HGUST,4HBLST,4HRI GD
1, 4HMERG, 4H /, 4H /, NOPT/8/

C
NIN = 5
NCUT = 5
KROW = 54
IPRNT = 1
LINES = 0

C
NTI = 31
NP2 = 2

C
DDTBL(1,1) = 19
DDTBL(2,1) = 4HAERO
DDTBL(3,1) = 2
DDTBL(1,5) = 35
DDTBL(2,5) = 4HFERSP
DDTBL(3,5) = 2
DDTBL(1,4) = 34
DDTBL(2,4) = 4HUNIT
DDTBL(3,4) = 2
DDTBL(1,6) = 36
DDTBL(2,6) = 4HLOAD
DDTBL(3,6) = 2

C
DO 10 I=1,100
10 NF(I) = 0

C
NAA = 1
TUNITS=2

C
HERE STARTS THE RUN DATA DECK INPUT

C
READ(NIN,300)IDENT,KPRCXQ,KPRCHQ,KPRCXL,KPRCHL

C
IF (IDENT.LE.0) GO TO 290
CALL HEADNG
WRITE(NOUT,320)IDENT

C
READ(NIN,305)(OPDATA(I),I=1,6)

C
305 FORMAT(5(A4,4X))

```

```

75 DC 75 I=1,NOPT
   OPDATA(I)=BLANK
   DO 80 I=1,6
   DC 80 J=1,NOPT
   IF (OPDATA(I).EQ.OPT(J)) OPDATA(J) = OPT(J)
80 CONTINUE
C
600 WRITE (NOUT,600) (OPDATA(I),I=1,NOPT)
600 FORMAT(1H0,20X, THE FOLLOWING ANALYSIS CODES HAVE BEEN .
1 *CALLED FOR IN THIS RUN*/1H0,20X,10(A4,4X))
C
   IPRNTM=0
   DC 85 I=2,4
   IF (OPDATA(I).NE.BLANK) IPRNTM=1
C
   IF (OPDATA(8).NE.BLANK) GO TO 280
   IF (OPDATA(1).EQ.BLANK) GO TO 105
C
90 FORMAT (A4)
C
CALL CSOLM (NTI)
C
DC 101 I=2,7
IF (OPDATA(I).NE.BLANK) GO TO 105
101 CONTINUE
GO TO 290
C
105 CALL RDAERO (4,HHEAD,0,IHD,NOUT,NEK)
C
   NER EQUALS 0 MEANS NO END OF FILE AND AERO ALREADY EXISTS
   IF (NER.EQ.0) GO TO 110
C
   WRITE (NOUT,502)
602 FORMAT (1H1,20X, AERO FILE IS NOT DEFINED - STOP )
GO TO 290
C
110 NK = IHD(2)
   NK0 = IHD(2)
   NSYM = IHD(3)
   NASYM = IHD(4)
   NG = IHD(5)
   NR = IHD(5)
   NROX = IHD(7)
   NSBETO = IHD(8)
   BZERO = RHD(24)/2.0
C
   READ(NIN,310)ALT,VKEAS,SIZECT,AIPLQ,AIPLL
   IPLQ=AIPLQ
   IPLL=AIPLL
   IF (SIZECT.EQ.0) SIZECT=1.0
C
   SIZECT IS 1.0 IF ALL GEOM DATA INPUT IN INCHES
   IF NOT, THEN SIZECT=SIZE IN INCHES/SIZE IN YOUR UNITS
C
C
C

```

DACGUL06
 DACGUL07
 DACGUL08
 DACGUL09
 DACGUL10
 DACGUL11
 DACGUL12
 DACGUL13
 DACGUL14
 DACGUL15
 DACGUL16
 DACGUL17
 DACGUL18
 DACGUL19
 DACGUL20
 DACGUL21
 DACGUL22
 DACGUL23
 DACGUL24
 DACGUL25
 DACGUL26
 DACGUL27
 DACGUL28
 DACGUL29
 DACGUL30
 DACGUL31
 DACGUL32
 DACGUL33
 DACGUL34
 DACGUL35
 DACGUL36
 DACGUL37
 DACGUL38
 DACGUL39
 DACGUL40
 DACGUL41
 DACGUL42
 DACGUL43
 DACGUL44
 DACGUL45
 DACGUL46
 DACGUL47
 DACGUL48
 DACGUL49
 DACGUL50
 DACGUL51
 DACGUL52
 DACGUL53
 DACGUL54
 DACGUL55
 DACGUL56
 DACGUL57

```

C BUT REMEMBER MASS IS MASS
C IF (VKEAS.LE.0) STOP
C CALL ATMDS(ALT,TEMP,SIGMA,RHO,TR,PO,VSS,AMU,KK)
RHO=RHO/SIGMA
EQUVAS=SQRT(SIGMA)
VTFPS=VKEAS/(UNITS(IUNITS)*EQUVAS)
DYNP=0.50*RHO*VTFPS**2
PO=14.696*PO
AMACH=VTFPS/VSS
CALL HEADNG
WRITE(INOUT,360)VKEAS,ALT,SIZECT,VTFPS,DYNP,SIGMA,AMACH
1, IPLQ,IPL
C IF (AMACH.LT.1.0) GO TO 200
WRITE(INOUT,770)
FORMAT(1H1,20X, OH OH MACH IS OVER 1.0 STOP )
770 STOP
C 200 CONTINUE
C IF (OPDATA(7).NE.BLANK) GO TO 270
C FIND THE I MOD OR SECT INPUT DATA
C 120 READ (INTI,90) CHECK
IF (CHECK.NE.AMOD) GO TO 122
I SECT = 0
IMS = 1
NDOF = 3
C I MOD INPUT DATA FOUND
C READ (INTI,300) NMS, NENG, KPRLDS
WRITE(INOUT,370) NMS, NSYM, NASYM, NENG, KPRLDS
FORMAT(1H0,20X, INERTIAL DATA /1H0,
1 20X, NO. TOTAL MASSES = ,14/1H,
1 20X, NO. SYMMETRIC MODES = ,14/1H,
1 20X, NO. ANTI SYMMETRIC MODES = ,14/1H,
1 20X, NO. ENGINES = ,14/1H,
1 20X, UNIT LOAD PRNT FLAG = ,14/1H,
GO TO 125
C 122 IF (CHECK.NE.SECT) GO TO 120
I SECT = 1
READ (INTI,300) NMS, NDOF, MSYM, NASYM, NENG, KPRLDS
C *** CHECK MSYM AND NASYM
C IF (NSYM.EQ.MSYM.AND.NASYM.EQ.MASYM) GO TO 124
NFP = 1

```

DACGU158
 DACGU159
 DACGU160
 DACGU161
 DACGU162
 DACGU163
 DACGU164
 DACGU165
 DACGU166
 DACGU167
 DACGU168
 DACGU169
 DACGU170
 DACGU171
 DACGU172
 DACGU173
 DACGU174
 DACGU175
 DACGU176
 DACGU177
 DACGU178
 DACGU179
 DACGU180
 DACGU181
 DACGU182
 DACGU183
 DACGU184
 DACGU185
 DACGU186
 DACGU187
 DACGU188
 DACGU189
 DACGU190
 DACGU191
 DACGU192
 DACGU193
 DACGU194
 DACGU195
 DACGU196
 DACGU197
 DACGU198
 DACGU199
 DACGU200
 DACGU201
 DACGU202
 DACGU203
 DACGU204
 DACGU205
 DACGU206
 DACGU207
 DACGU208
 DACGU209

```

WRITE (NOUT,123)
123 ECPMAT (140,*NSYM AND/OR NASYM INCOMPATIBLE WITH THE AERO FILE*)
   GO TO 290
C
124 IMS = 10
   IF (NDOF.EQ.7) IMS = 16
   IF (NDOF.EQ.8) IMS = 23
125 CONTINUE
C
NCW READ IN ACS DATA IF RQD
C
NTFS=0
NTFA=0
MXBLK=0
MXORLK=0
MXORD=0
C
IF (OPDATA(3).EQ.BLANK)GO TO 205
PEAD(NIN,300)NTFS,NTFA,MXBLK,MXORLK
C
WRITE (NOUT,511)NTFS,NTFA,MXBLK,MXORLK
C
MXORLK=MXORBLK+1
MXORD=MXORBLK+MXRLK
C
205 C(NTI NUF
C
NOMOD = NSYM+NASYM
IDIMUL=1
IF (OPDATA(2).EQ.BLANK)IDIMUL=0
C
L1 = 1
L2 = L1 + 20
L3 = L2 + NOMOD
L4 = L3 + NOMOD*NOMOD
L41 = L4 + NOMOD
L41NR = L41
L5 = L41 + NOMOD
L10 = L9 + NMS*NOMOD
L11 = L10 + NMS*NOMOD
LMODE = L11 + NMS*NOMOD
IF (NDOF.NE.3) LMODE = L9 + NDOF*NMS*NOMOD
L51 = LMODE
L52=L51+NTFS*4
L53=L52+NTFA*4
L54=L53+2*NTFS*MXORD
L55=L54+2*NTFA*MXORD
L56=L55+NTFS*NSYM
LFRQS = L56 + NTFA*NASYM
L5 = LFRQS
L6 = L5 + IMS*NMS

```

DACGU210
 DACGU211
 DACGU212
 DACGU213
 DACGU214
 DACGU215
 DACGU216
 DACGU217
 DACGU218
 DACGU219
 DACGU220
 DACGU221
 DACGU222
 DACGU223
 DACGU224
 DACGU225
 DACGU226
 DACGU227
 DACGU228
 DACGU229
 DACGU230
 DACGU231
 DACGU232
 DACGU233
 DACGU234
 DACGU235
 DACGU236
 DACGU237
 DACGU238
 DACGU239
 DACGU240
 DACGU241
 DACGU242
 DACGU243
 DACGU244
 DACGU245
 DACGU246
 DACGU247
 DACGU248
 DACGU249
 DACGU250
 DACGU251
 DACGU252
 DACGU253
 DACGU254
 DACGU255
 DACGU256
 DACGU257
 DACGU258
 DACGU259
 DACGU260
 DACGU261

DACGU262
 DACGU263
 DACGU264
 DACGU265
 DACGU266
 DACGU267
 DACGU268
 DACGU269
 DACGU270
 DACGU271
 DACGU272
 DACGU273
 DACGU274
 DACGU275
 DACGU276
 DACGU277
 DACGU278
 DACGU279
 DACGU280
 DACGU281
 DACGU282
 DACGU283
 DACGU284
 DACGU285
 DACGU286
 DACGU287
 DACGU288
 DACGU289
 DACGU290
 DACGU291
 DACGU292
 DACGU293
 DACGU294
 DACGU295
 DACGU296
 DACGU297
 DACGU298
 DACGU299
 DACGU300
 DACGU301
 DACGU302
 DACGU303
 DACGU304
 DACGU305
 DACGU306
 DACGU307
 DACGU308
 DACGU309
 DACGU310
 DACGU311
 DACGU312
 DACGU313

```

L7 = L6 + NMS
L8 = L7 + NMS
L12 = L8 + NMS

L13 = L12 + NDOF*NMS*NMOD*IDIMUL
L14 = L13 + 2*NENGS*IDIMUL
L15 = L14 + NDOF*NDOF*NMS*ISECT
NAA = L15
WRITE(NOUT,999)NAA,CDPERQ

CALL INTRM(NTI,NMS,
1 A(L5),A(L6),A(L7),A(L8),A(L9),A(L10),A(L11),
2 A(L14),
1 A(L1),A(L41),A(L12),A(L3),A(L4),NMOD,NSYM,NASYM)

CALL HEADNG
IF(OPDATA(2).EQ.BLANK)GO TO 207
FIND THE LOAD INPUT DATA

130 READ (NTI,90) CHECK
IF (CHECK.NE.AL0D) GO TO 130
LOAD INPUT DATA FOUND

READ IN HERE THE CONTROLLING DATA FOR UNIT LOADS
READ (NTI,300) NBEAMS,NINTLD,NSTRSS,NMGRP,NAHGRP,NSHGRP
WRITE(NOUT,345)NBFAMS,NINTLD,NSTRSS,NMGRP,NAHGRP,NSHGRP

345 FORMAT(1H0,20X,UNIT LOAD DATA /1H0,
1 20X,N0,INTGD LOAD BEAMS = ,14/1H,
1 20X,N0,INTGD LOADS DEFINED = ,14/1H,
1 20X,N0,STRESSES = ,14/1H,
1 20X,N0,MASS GROUPS = ,14/1H,
1 20X,N0,AERO BOX GROUPS = ,14/1H,
1 20X,N0,SLENDER BODY GROUPS = ,14)

CALL LOAD (NTI, A(L6), A(L7), A(L8), A(L12), NDOF,NMS,NSYM,NASYM,
1 A(L9),A(L10),A(L11),NENGS,A(L13))
IF (NER.NE.0) GO TO 290

GO TO 220

207 IF (OPDATA(5).EQ.BLANK.AND.OPDATA(6).EQ.BLANK) GO TO 220

CALL ROUNIT (1,0,1H0,NOUT,NER)
NER EQUALS 0 MEANS NO END OF FILE AND UNIT ALREADY EXISTS
IF (NER.EQ.0) GO TO 210

```

```

603      WRITE (NOUT,603)
      GO TO 290
C
210      NIN1D = IHD(3)
      NSTRSS = IHD(4)
      NENG5 = IHD(5)
      NAGGRP = IHD(9)
      NS4GRP = IHD(10)
      NPROXS = IHD(11)
      NAERSB = IHD(12)
C
220      CONTINUE
C
      IF (CPDATA(3).EQ.BLANK) GO TO 225
      IF (NTES.NE.0.OR.NTFA.NE.0) CALL ACS(
1      NIN,MAXBLK,MXCHLK,A(L9),A(L10),A(L11),
1      A(L51),A(L52),A(L53),A(L54),A(L55),A(L56))
C
225      CONTINUE
C
      IF (OPDATA(4).EQ.BLANK.AND.OPDATA(5).EQ.BLANK.AND.
1      OPDATA(6).EQ.BLANK) GO TO 290
C
      NAA = LFRORS
C
      NTOTAP=NRDX+2*NSBETN
      MTOTAP=2*NTOTAP
C
      L60 = NAA
      L61 = L60
      LENGTH=MAXO(2*NSYM*NSYM,2*NASYM*NASYM,NTOTAP*(NSYM+NASYM),
1      2*NTOTAP*NG)
      L62 = L61 + LENGTH
      L63 = L62
      L64 = L63 + NTOTAP*NSYM
      LTAERO = L64 + NTOTAP*NASYM
C
      WRITE (NOUT,999) LTAERC,CCRERA
C
C
      CALL TRAERO (NOUT,NER,MTOTAP,NTOTAP
1      ,CR,A(L60),A(L61),A(L62),A(L63),A(L64) )
      IF (NER.NE.0) GO TO 290
C
      IF (OPDATA(4).EQ.BLANK) GO TO 227
C
      NOW READ IN FREQS FOR FREQ RESPONSE SOLN IF RQD
      READ(NIN,300)NFRGR
      I = 0

```

```

DACGU314
DACGU315
DACGU316
DACGU317
DACGU318
DACGU319
DACGU320
DACGU321
DACGU322
DACGU323
DACGU324
DACGU325
DACGU326
DACGU327
DACGU328
DACGU329
DACGU330
DACGU331
DACGU332
DACGU333
DACGU334
DACGU335
DACGU336
DACGU337
DACGU338
DACGU339
DACGU340
DACGU341
DACGU342
DACGU343
DACGU344
DACGU345
DACGU346
DACGU347
DACGU348
DACGU349
DACGU350
DACGU351
DACGU352
DACGU353
DACGU354
DACGU355
DACGU356
DACGU357
DACGU358
DACGU359
DACGU360
DACGU361
DACGU362
DACGU363
DACGU364
DACGU365

```

```

DC 150 N=1,NFRGR
READ (NIN,310) (RINTP(N,J),J=1,3)
I = I+1
VGBWIN(I) = RINTP(N,1)
140 I = I+1
VGBWIN(I) = VGBWIN(I-1) + RINTP(N,3)
IF (VGBWIN(I).LT.RINTP(N,2)) GO TO 140
IF (VGBWIN(I).GT.RINTP(N,2)) I=I-1
150 NREQ = I
C
WRITE (NOUT,30) NREQ
DC 155 I=1,NFRGR
155 WRITE (NOUT,31) I, (RINTP(I,J),J=1,3)
C
30 FORMAT(1H0,20X, SOLUTION FREQUENCY REQUEST FOR ,I4, FREQUENCIES
1 1H0, 1TM ,6X, LWR F ,6X, UPR F ,6X, DEL F )
31 FORMAT(1H ,I3,3(1X,F10.4))
C
KPRCXI=KPRCXQ
KPRCHK=KPRCHKQ
C
CALL CFREQR (CR,A(11),A(12),A(13),A(14),A(141),N3MOD,NREQ,NG,
1 A(151),A(152),A(153),A(154),A(155),A(156)
2 )
C
227 IF (OPDATA(5).EQ.BLANK) GO TO 247
C
CALL RDRSP (1,0,0,IHD,NOUT,NFR)
NFR EQUALS 0 MEANS NO END OF FILE AND FRSP ALREADY EXISTS
IF (NFR.EQ.0) GO TO 230
C
WRITE (NOUT,604)
604 FORMAT (1H1,20X, FRSP FILE IS NOT DEFINED - STOP )
GO TO 290
C
230 NREQ = IHD(4)
NK = IHD(7)
VTEPS = RHD(11)
SIGMA = RHD(12)
C
READ (NIN,300) NACC
NAA=LFRQKS
C
KPRCXI=KPRCXL
KPRCHK=KPRCHKL
C
CALL CFRLND(CR,A(11),A(111),NMS,
1 A(14),A(110),A(111),NMS,
1 A(151),A(152),A(153),A(154),A(155),A(156))
C
247 IF (OPDATA(6).EQ.BLANK) GO TO 290

```

DACGU366
 DACGU367
 DACGU368
 DACGU369
 DACGU370
 DACGU371
 DACGU372
 DACGU373
 DACGU374
 DACGU375
 DACGU376
 DACGU377
 DACGU378
 DACGU379
 DACGU380
 DACGU381
 DACGU382
 DACGU383
 DACGU384
 DACGU385
 DACGU386
 DACGU387
 DACGU388
 DACGU389
 DACGU390
 DACGU391
 DACGU392
 DACGU393
 DACGU394
 DACGU395
 DACGU396
 DACGU397
 DACGU398
 DACGU399
 DACGU400
 DACGU401
 DACGU402
 DACGU403
 DACGU404
 DACGU405
 DACGU406
 DACGU407
 DACGU408
 DACGU409
 DACGU410
 DACGU411
 DACGU412
 DACGU413
 DACGU414
 DACGU415
 DACGU416
 DACGU417

DACGU418
 DACGU419
 DACGU420
 DACGU421
 DACGU422
 DACGU423
 DACGU424
 DACGU425
 DACGU426
 DACGU427
 DACGU428
 DACGU429
 DACGU430
 DACGU431
 DACGU432
 DACGU433
 DACGU434
 DACGU435
 DACGU436
 DACGU437
 DACGU438
 DACGU439
 DACGU440
 DACGU441
 DACGU442
 DACGU443
 DACGU444
 DACGU445
 DACGU446
 DACGU447
 DACGU448
 DACGU449
 DACGU450
 DACGU451
 DACGU452
 DACGU453
 DACGU454
 DACGU455
 DACGU456
 DACGU457
 DACGU458
 DACGU459
 DACGU460
 DACGU461
 DACGU462
 DACGU463
 DACGU464
 DACGU465
 DACGU466
 DACGU467
 DACGU468
 DACGU469

```

C      CALL RDLQAD (1,0,IHD,NOUT,NER)
C      NER EQUALS 0 MEANS NO END OF FILE AND LOAD ALREADY EXISTS
C      IF (NER.EQ.0) GO TO 250
C
C      WRITE (NOUT,605)
C      605 FORMAT (1H1,20X, 'LOAD FILE IS NOT DEFINED - STOP ')
C      GO TO 290
C
C      250 NFRQ = IHD(4)
C      NINTLO = IHD(5)
C      NACC = IHD(6)
C
C      KPRCH=K=0
C      KPRCXI=0
C      NAA = LMINB
C      IF PR RUT MODES ARE FOR AN ANGLE OTHER THAN 1 RADIAN INPUT
C      PATIO OF 1 RADIAN/RADIANS USED FOR TRIM SOLN CORRECTION
C
C      READ(NIN,310)AN,AKMAN,ZDOT,AKPRCH,RBRADF
C      KMAN=AKMAN
C      KPTRM=AKPRCH
C
C      READ(NIN,310)TIMEMX,DELT,EFR,HGRD,AKPRTM,AIPLBL
C      IPLBL=AIPLBL
C      KPRTMH=AKPRTM
C
C      READ(NIN,300)NORMAX,KGRD,KLPT,KLOAD,NCRITS,KPRRLS
C
C      CALL HEADNG
C      WRITE(NOUT,60)AN,KMAN,ZDOT,KPRTPM,RBRADF
C
C      WRITE(NOUT,70)TIMEMX,EFR,HGRD,KPRTMH,NORMAX,KGRD,KLPT,KLOAD,
C      1 ACRTS,KPRBS,IPLBL,DELT
C
C      CALL CGUST (CP,A(11),A(13),A(14) )
C
C      GO TO 290
C
C      270 CALL CRIGID
C      GO TO 290
C
C      280 CALL CMERGE
C
C      290 STOP
C
C      300 FCRMAT (6112)
C      310 FCRMAT (6112,1)
C      320 FCRMAT(1H0,20X, 'FIXED DATA DECK INPUT GO /1H0,20X, IDENT= ,112)
C      360 FCRMAT(1H0,20X, 'RIN DATA DECK INPUT GO /1H0,

```

```

1 20X, FLIGHT DATA = ,F8.3, KEAS /1H, DACGU470
1 20X, VKFAS = ,F10.3, FT. /1H, DACGU471
1 20X, ALTITUDE = ,F8.4, INCHES/UNIT /1H, DACGU472
1 20X, SIZE FACTOR = ,F8.3, FPS /1H, DACGU473
1 20X, VTRUE = ,F8.3, PSF /1H, DACGU474
1 20X, DYNAMIC P = ,F12.5/1H, DACGU475
1 20X, SIGMA = ,F8.4/1H, DACGU476
1 20X, MACH = ,F14/1H, DACGU477
1 20X, PLOT Q FLG = ,F14, DACGU478
511 FORMAT(1H0,20X, ACTIVE CONTROL SYSTEM DATA /1H0,
1 20X, NTES = ,F14/1H, DACGU479
1 20X, NTFA = ,F14/1H, DACGU480
1 20X, MYRLK = ,F14/1H, DACGU481
1 20X, MXDBLK = ,F14, DACGU482
990 FORMAT(1H0,20X,110, WORDS OF CORE REQ FOR STEP +++,A10, +++ )
K0 FORMAT(1H0,20X, RUN DATA FOR TRIM /1H0,
1 20X, LOAD FACTOR = ,F8.3/1H, DACGU484
1 20X, MANEUVER CODE = ,F14 /1H, DACGU485
1 20X, ZFFT = ,F8.3, FPS /1H, DACGU486
1 20X, TRIM MIX PRNT FLG = ,F14/1H, DACGU487
1 20X, RB MODE ROT. CORR. = ,F12.5, DACGU488
70 FORMAT(1H0,20X, RUN DATA FOR BLAST /1H0,
1 20X, YIELD = ,F8.3, SECS /1H,
1 20X, MAX SOLN TIME = ,F8.3, KT /1H,
1 20X, GROUND HEIGHT = ,F8.1, FT. /1H,
1 20X, NO. ORIENTATIONS = ,F14 /1H,
1 20X, GROUND PEFL. CODE = ,F14 /1H,
1 20X, ITERATION CODE = ,F14 /1H,
1 20X, MOD CRIT LOAD CODE = ,F14 /1H,
1 20X, MOD CRIT STRESS CODE = ,F14 /1H,
1 20X, LOAD MIX PRNT FLG = ,F14/1H,
1 20X, PLST BLST LOAD FLG = ,F10.5, SECS )
1 20X, INITIAL DELTA T = ,F10.5, SECS )
FND

```

```

SUBROUTINE ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
CALLING SEQUENCE
CALL ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
Z = GEOMETRIC ALTITUDE (FT)
TM = MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE)
THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL
CA = SPEED OF SOUND (FT/SEC)
AMU = VISCOSITY COEFFICIENT (LB-SEC/FT**2)
K = 1 NORMAL, GREATER THAN 300000. FT.,
    = 2 ALTITUDE NEGATIVE,
    = 3 ALTITUDE GREATER THAN 300000. FT.,
    = 4 FLOATING POINT OVERFLOW,
    = 5 ALTITUDE GREATER THAN 300000. FT. AND FLOATING POINT OVERFLOW.
DIMENSION HPRI MB(11), TMB(11), SIGMA B(11), ALM(11)
DATA HPRI MB/0.0,36099.239,82020.997,154199.480,173884.510,259186.3ATMOS
150,295275.590,344488.190,524934.380,557742.780,656167.980,718/518ATMOS
2.688,389.088,389.988,508.788,298.188,406.188,2386.ATMOS
3188,2566.188,2836.188/,SIGMA B/1.00,2.9706958E-01,3.2665751E-02,1.2ATMOS
4117870E-03,5.8677311E-04,1.7329156E-05,1.7928595E-06,9.3921519E-08ATMOS
5,7.7658593E-10,5.6324877E-10,2.5726771E-10/,ALM/-0.00356616,0.0,0.0ATMOS
60164592.0,0,-0.00246888,0.0,0.00219456,0.01097280,0.00548640,0.000ATMOS
7274320.0,0.00192024/,RE/2.0855531E07/,S/198.72/,AMU/3.7372999E-07/ATMOS
1,RHO/0.0023769/,TM/518.688/ATMOS
K=1
IF (Z) 10,30,20
K=3
GO TO 110
IF (Z.GT.300000.) K=K+1
HPRI MB=(RHO/(RE+Z))**Z
GO 40 M=1,11
IF (HPRI MB-HPRI MB(M)) 50,60,40
CONTINUE
M=12
M=M-1
IF (ALM(M)) 70,80,70
TM=TM*(M)+ALM(M)*(HPRI MB-HPRI MB(M))
SIGMA B=EXP((1.0+(Q/ALM(M)))*(LOG(TMB(M)/TM)))*SIGMA B(M)
GO TO 90
TM=TMB(M)
SIGMA B=SIGMA B(M)*EXP(-(Q*(HPRI MB-HPRI MB(M)))/TM*(M))
RHO=RHO**SIGMA B
THETA=TM/TMZ
DELTA=SIGMA B*THETA
CA=49.02177*SQRT(TM)
AMU=AMU**SQRT(THETA**3)*((TMZ+S)/(TM+S))
K=K+3
RETURN
END

```

0000000000000000

10
20
30
40
50
60
70
80
90
100
110

```

SUBROUTINE FDDTBL (CH,JI,J0)
COMMON /DDTALS/DDTBL(20,10)
INTEGER DDTBL, CH
JI = 0
J0 = 0
DO 100 J=1,10
IF (DDTBL(2,J).NE.CH) GO TO 100
IF (DDTBL(3,J).NE.3) JI=J
IF (DDTBL(3,J).NE.1) J0=J
100 CONTINUE
RETURN
END

```

```

FDDTBL 2
FDDTBL 3
FDDTBL 4
FDDTBL 5
FDDTBL 6
FDDTBL 7
FDDTBL 8
FDDTBL 9
FDDTBL 10
FDDTBL 11
FDDTBL 12
FDDTBL 13

```

HEADNG 2
 HEADNG 3
 HEADNG 4
 HEADNG 5
 HEADNG 6
 HEADNG 7
 HEADNG 8

SUBROUTINE HEADNG
 COMMON /ZZZ/CASE(30),X(12),DATE(2),TIME(2),Y,Z,NIN,NOUT,W,LINES
 FORMAT (1H1,T8,4HDATE,2X2A4,T2,15A4/1X15A4,T8,4HTIME,2X2A4)
 WRITE (NCUT,1) DATE,CASE,TIME
 LINES = 2
 RETURN
 END

1

[illegible]

```

54 NEFP = 2
55 GO TO 190
56
57 START REDUCTION OF MATRIX A
58
59 4) DO 130 I = 1, N
60
61 SEARCH FOR MAXIMUM ELEMENT IN ITH ROW OF A-MATRIX
62
63 AIJMAX = A(I,1)
64 JMAX = 1
65 DO 50 J = 2, N
66 IF ( ABS( A(I,J) ) -LE. ABS( AIJMAX ) ) GO TO 50
67 AIJMAX = A(I,J)
68 JMAX = J
69 CONTINUE
70
71 IF AIJMAX IS ZERO, THE MATRIX IS SINGULAR
72
73 IF ( ABS( AIJMAX ) .GT. 0.0EO ) GO TO 70
74
75 6) NEFP = 1
76 GO TO 190
77
78 NORMALIZE ITH ROW BY AIJMAX (JMAX ELEMENT OF ITH ROW)
79
80 DO 80 J = 1, N
81 A(I,J) = A(I,J) / AIJMAX
82 D = D * AIJMAX
83
84 NORMALIZE ITH ROW OF B
85
86 DO 90 J = 1, MD
87 B(I,J) = B(I,J) / AIJMAX
88
89 USE ROW TRANSFORMATIONS TO GET ZEROS ABOVE AND BELOW THE JMAX
90 ELEMENT OF THE ITH ROW OF A. APPLY SAME ROW TRANSFORMATIONS
91 TO THE B MATRIX.
92
93 DO 120 K = 1, N GO TO 120
94 IF ( K.EQ. I )
95 APAT = -A(K,JMAX)
96 DO 100 J = 1, N
97 IF ( ABS( A(I,J) ) -EQ. 0.0EO ) GO TO 100
98 A(K,J) = APAT * A(I,J) + A(K,J)
99 CONTINUE
100 A(K,JMAX) = 0.0EO
101 DO 110 J = 1, MD
102 IF ( ABS( B(I,J) ) -EQ. 0.0EO ) GO TO 110
103 B(K,J) = APAT * B(I,J) + B(K,J)
104 CONTINUE
105
106 110 CONTINUE
107
108 120 CONTINUE

```

MISI 106
 MISI 107
 MISI 108
 MISI 109
 MISI 110
 MISI 111
 MISI 112
 MISI 113
 MISI 114
 MISI 115
 MISI 116
 MISI 117
 MISI 118
 MISI 119
 MISI 120
 MISI 121
 MISI 122
 MISI 123
 MISI 124
 MISI 125
 MISI 126
 MISI 127
 MISI 128
 MISI 129
 MISI 130
 MISI 131
 MISI 132
 MISI 133
 MISI 134
 MISI 135
 MISI 136
 MISI 137
 MISI 138

```

C      STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN.  THUS,
C      THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITV)
C      ELEMENT OF EACH COLUMN (AFTER REDUCTION).
C
C      L = I
C      130 A(I,JMAX) = EL      THIS STORES INTEGER IN TOP ROW OF A
C
C      THE REDUCTION OF A IS NOW COMPLETE.  PERFORM ROW INTERCHANGES
C      AS INDICATED IN THE FIRST ROW OF A.
C
C      DO 170 I = 1, N
C      K = I
C      140 FK = A(I,K)      THIS PUTS THE INTEGER VALUE IN A INTO K
C      IF (K - I)140,170,150
C
C      IF K(I,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN
C      INVOLVED IN AN INTERCHANGE, AND WE USE K(I,K) UNTIL WE GET
C      A VALUE OF K GREATER THAN I (CORRESPONDING TO A ROW STORED
C      BELOW THE ITH ROW).
C
C      150 DO 160 J = 1, MD
C      A(I,J) = A(K,J)
C      160 R(K,J) = A(I,J)
C      D = -D
C      170 CONTINUE
C      180 NERR = 0
C      190 RETURN
C      END
  
```


MI	S2	54
MI	S2	55
MI	S2	56
MI	S2	57
MI	S2	58
MI	S2	59
MI	S2	60
MI	S2	61
MI	S2	62
MI	S2	63
MI	S2	64
MI	S2	65
MI	S2	66
MI	S2	67
MI	S2	68
MI	S2	69
MI	S2	70
MI	S2	71
MI	S2	72
MI	S2	73
MI	S2	74
MI	S2	75
MI	S2	76
MI	S2	77
MI	S2	78
MI	S2	79
MI	S2	80
MI	S2	81
MI	S2	82
MI	S2	83
MI	S2	84
MI	S2	85
MI	S2	86
MI	S2	87
MI	S2	88
MI	S2	89
MI	S2	90
MI	S2	91
MI	S2	92
MI	S2	93
MI	S2	94
MI	S2	95
MI	S2	96
MI	S2	97
MI	S2	98
MI	S2	99
MI	S2	100
MI	S2	101
MI	S2	102
MI	S2	103
MI	S2	104
MI	S2	105

MI S2 106
MI S2 107
MI S2 108
MI S2 109
MI S2 110
MI S2 111
MI S2 112
MI S2 113
MI S2 114
MI S2 115
MI S2 116
MI S2 117
MI S2 118
MI S2 119
MI S2 120
MI S2 121
MI S2 122
MI S2 123
MI S2 124
MI S2 125
MI S2 126
MI S2 127
MI S2 128
MI S2 129
MI S2 130
MI S2 131
MI S2 132
MI S2 133
MI S2 134
MI S2 135
MI S2 136
MI S2 137
MI S2 138
MI S2 139
MI S2 140
MI S2 141

```

110 R(K,J) = ARAT * R(I,J) + B(K,J)
120 CONTINUE
130 STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN. THUS,
    THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITY)
    ELEMENT OF EACH COLUMN (AFTER REDUCTION).
    L = I
    130 A(I,JMAX) = FL
    THIS STORES INTEGER IN TOP ROW OF A
    THE REDUCTION OF A IS NOW COMPLETE. PERFORM ROW INTERCHANGES
    AS INDICATED IN THE FIRST ROW OF A.
    140 IF (I - 1) 140,170,150
    IF (I - 1) 140,170,150
    THIS PUTS THE INTEGER VALUE IN A INTO K
    K = I
    140 FK = A(I,K)
    IF (K - I) 140,170,150
    IF K(I,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN
    INVOLVED IN AN INTERCHANGE, AND WE USE K(I,K) UNTIL WE GET
    A VALUE OF K GREATER THAN I (CORRESPONDING TO A ROW STORED
    BELOW THE ITH ROW).
    150 DO 160 J = 1, MD
    ARAT = B(I,J)
    B(I,J) = R(K,J)
    160 B(K,J) = ARAT
    D = -D
    170 CONTINUE
    180 NEXT K = 0
    190 RETURN
    END

```

```

SUBROUTINE PLTI (N,X,Y,M)
PLTI - LINEAR PLOTTING ROUTINE
THIS SUBROUTINE GENERATES PRINTED OUTPUT IN GRAPHIC FORM. BOTH
AXES ARE GRADUATED IN A LINEAR MANNER.

      ***** CDC FTM VERSION OPT=1 *****
      N      = INPUT = NUMBER OF POINTS TO BE PLOTTED (INTEGER)
      X      = INPUT = LINEAR ARRAY (OF LENGTH N) CONTAINING THE
                  X-COORDINATES
      Y      = INPUT = LINEAR ARRAY (OF LENGTH N) CONTAINING THE
                  Y-COORDINATES
      M      = INPUT = NUMBER (INTEGER) OF 50 LINE PAGES THE PLOT
                  OCCUPIES IN Y-DIRECTION

FUNCTIONS USED - ABS, FLOOR
PRINTED OUTPUT IS PRODUCED ON UNIT 6 AFTER EJECTING A PAGE
DIMENSION FMT(24), FMTT(11), HH(11), FHH(8), FHL(8)
DIMENSION X(1), Y(1), KJ(1), ME(4), IRUF(101), S(11)
DATA ME(1) / 1H /, ME(2) / 1H /, ME(3) / 1H* /, ME(4) / 1H /
DATA FMT(1), FMT(24) / 4H(8X, 4H) /, FMTT(1), FMTT(2) / 4H(1XA, 4H) /
1 /, FMTT(5), FMTT(6), FMTT(7), FMTT(8), FMTT(11) / 4H(1H, 4H, 101, 4H) /
2 / 4H(1H, 4H) /, HH / 4H(0, 4H(1, 4H(2, 4H(3, 4H(4, 4H(5,
3 / 4H(6, 4H(7, 4H(8, 4H(9, 4H(10, 4H(11, 4H(12, 4H(13, 4H(14, 4H(15,
4 / 4H(16, 4H(17, 4H(18, 4H(19, 4H(20, 4H(21, 4H(22, 4H(23, 4H(24, 4H(25,
5 / 4H(26, 4H(27, 4H(28, 4H(29, 4H(30, 4H(31, 4H(32, 4H(33, 4H(34, 4H(35,
6 / 4H(36, 4H(37, 4H(38, 4H(39, 4H(40, 4H(41, 4H(42, 4H(43, 4H(44, 4H(45,
7 / 4H(46, 4H(47, 4H(48, 4H(49, 4H(50, 4H(51, 4H(52, 4H(53, 4H(54, 4H(55,
8 / 4H(56, 4H(57, 4H(58, 4H(59, 4H(60, 4H(61, 4H(62, 4H(63, 4H(64, 4H(65,
9 / 4H(66, 4H(67, 4H(68, 4H(69, 4H(70, 4H(71, 4H(72, 4H(73, 4H(74, 4H(75,
10 / 4H(76, 4H(77, 4H(78, 4H(79, 4H(80, 4H(81, 4H(82, 4H(83, 4H(84, 4H(85,
11 / 4H(86, 4H(87, 4H(88, 4H(89, 4H(90, 4H(91, 4H(92, 4H(93, 4H(94, 4H(95,
12 / 4H(96, 4H(97, 4H(98, 4H(99, 4H(100, 4H(101, 4H(102, 4H(103, 4H(104,
13 / 4H(105, 4H(106, 4H(107, 4H(108, 4H(109, 4H(110, 4H(111, 4H(112, 4H(113,
14 / 4H(114, 4H(115, 4H(116, 4H(117, 4H(118, 4H(119, 4H(120, 4H(121, 4H(122,
15 / 4H(123, 4H(124, 4H(125, 4H(126, 4H(127, 4H(128, 4H(129, 4H(130, 4H(131,
16 / 4H(132, 4H(133, 4H(134, 4H(135, 4H(136, 4H(137, 4H(138, 4H(139, 4H(140,
17 / 4H(141, 4H(142, 4H(143, 4H(144, 4H(145, 4H(146, 4H(147, 4H(148, 4H(149,
18 / 4H(150, 4H(151, 4H(152, 4H(153, 4H(154, 4H(155, 4H(156, 4H(157, 4H(158,
19 / 4H(159, 4H(160, 4H(161, 4H(162, 4H(163, 4H(164, 4H(165, 4H(166, 4H(167,
20 / 4H(168, 4H(169, 4H(170, 4H(171, 4H(172, 4H(173, 4H(174, 4H(175, 4H(176,
21 / 4H(177, 4H(178, 4H(179, 4H(180, 4H(181, 4H(182, 4H(183, 4H(184, 4H(185,
22 / 4H(186, 4H(187, 4H(188, 4H(189, 4H(190, 4H(191, 4H(192, 4H(193, 4H(194,
23 / 4H(195, 4H(196, 4H(197, 4H(198, 4H(199, 4H(200, 4H(201, 4H(202, 4H(203,
24 / 4H(204, 4H(205, 4H(206, 4H(207, 4H(208, 4H(209, 4H(210, 4H(211, 4H(212,
25 / 4H(213, 4H(214, 4H(215, 4H(216, 4H(217, 4H(218, 4H(219, 4H(220, 4H(221,
26 / 4H(222, 4H(223, 4H(224, 4H(225, 4H(226, 4H(227, 4H(228, 4H(229, 4H(230,
27 / 4H(231, 4H(232, 4H(233, 4H(234, 4H(235, 4H(236, 4H(237, 4H(238, 4H(239,
28 / 4H(240, 4H(241, 4H(242, 4H(243, 4H(244, 4H(245, 4H(246, 4H(247, 4H(248,
29 / 4H(249, 4H(250, 4H(251, 4H(252, 4H(253, 4H(254, 4H(255, 4H(256, 4H(257,
30 / 4H(258, 4H(259, 4H(260, 4H(261, 4H(262, 4H(263, 4H(264, 4H(265, 4H(266,
31 / 4H(267, 4H(268, 4H(269, 4H(270, 4H(271, 4H(272, 4H(273, 4H(274, 4H(275,
32 / 4H(276, 4H(277, 4H(278, 4H(279, 4H(280, 4H(281, 4H(282, 4H(283, 4H(284,
33 / 4H(285, 4H(286, 4H(287, 4H(288, 4H(289, 4H(290, 4H(291, 4H(292, 4H(293,
34 / 4H(294, 4H(295, 4H(296, 4H(297, 4H(298, 4H(299, 4H(300, 4H(301, 4H(302,
35 / 4H(303, 4H(304, 4H(305, 4H(306, 4H(307, 4H(308, 4H(309, 4H(310, 4H(311,
36 / 4H(312, 4H(313, 4H(314, 4H(315, 4H(316, 4H(317, 4H(318, 4H(319, 4H(320,
37 / 4H(321, 4H(322, 4H(323, 4H(324, 4H(325, 4H(326, 4H(327, 4H(328, 4H(329,
38 / 4H(330, 4H(331, 4H(332, 4H(333, 4H(334, 4H(335, 4H(336, 4H(337, 4H(338,
39 / 4H(339, 4H(340, 4H(341, 4H(342, 4H(343, 4H(344, 4H(345, 4H(346, 4H(347,
40 / 4H(348, 4H(349, 4H(350, 4H(351, 4H(352, 4H(353, 4H(354, 4H(355, 4H(356,
41 / 4H(357, 4H(358, 4H(359, 4H(360, 4H(361, 4H(362, 4H(363, 4H(364, 4H(365,
42 / 4H(366, 4H(367, 4H(368, 4H(369, 4H(370, 4H(371, 4H(372, 4H(373, 4H(374,
43 / 4H(375, 4H(376, 4H(377, 4H(378, 4H(379, 4H(380, 4H(381, 4H(382, 4H(383,
44 / 4H(384, 4H(385, 4H(386, 4H(387, 4H(388, 4H(389, 4H(390, 4H(391, 4H(392,
45 / 4H(393, 4H(394, 4H(395, 4H(396, 4H(397, 4H(398, 4H(399, 4H(400, 4H(401,
46 / 4H(402, 4H(403, 4H(404, 4H(405, 4H(406, 4H(407, 4H(408, 4H(409, 4H(410,
47 / 4H(411, 4H(412, 4H(413, 4H(414, 4H(415, 4H(416, 4H(417, 4H(418, 4H(419,
48 / 4H(420, 4H(421, 4H(422, 4H(423, 4H(424, 4H(425, 4H(426, 4H(427, 4H(428,
49 / 4H(429, 4H(430, 4H(431, 4H(432, 4H(433, 4H(434, 4H(435, 4H(436, 4H(437,
50 / 4H(438, 4H(439, 4H(440, 4H(441, 4H(442, 4H(443, 4H(444, 4H(445, 4H(446,
51 / 4H(447, 4H(448, 4H(449, 4H(450, 4H(451, 4H(452, 4H(453, 4H(454, 4H(455,
52 / 4H(456, 4H(457, 4H(458, 4H(459, 4H(460, 4H(461, 4H(462, 4H(463, 4H(464,
53 / 4H(465, 4H(466, 4H(467, 4H(468, 4H(469, 4H(470, 4H(471, 4H(472, 4H(473,
54 / 4H(474, 4H(475, 4H(476, 4H(477, 4H(478, 4H(479, 4H(480, 4H(481, 4H(482,
55 / 4H(483, 4H(484, 4H(485, 4H(486, 4H(487, 4H(488, 4H(489, 4H(490, 4H(491,
56 / 4H(492, 4H(493, 4H(494, 4H(495, 4H(496, 4H(497, 4H(498, 4H(499, 4H(500,
57 / 4H(501, 4H(502, 4H(503, 4H(504, 4H(505, 4H(506, 4H(507, 4H(508, 4H(509,
58 / 4H(510, 4H(511, 4H(512, 4H(513, 4H(514, 4H(515, 4H(516, 4H(517, 4H(518,
59 / 4H(519, 4H(520, 4H(521, 4H(522, 4H(523, 4H(524, 4H(525, 4H(526, 4H(527,
60 / 4H(528, 4H(529, 4H(530, 4H(531, 4H(532, 4H(533, 4H(534, 4H(535, 4H(536,
61 / 4H(537, 4H(538, 4H(539, 4H(540, 4H(541, 4H(542, 4H(543, 4H(544, 4H(545,
62 / 4H(546, 4H(547, 4H(548, 4H(549, 4H(550, 4H(551, 4H(552, 4H(553, 4H(554,
63 / 4H(555, 4H(556, 4H(557, 4H(558, 4H(559, 4H(560, 4H(561, 4H(562, 4H(563,
64 / 4H(564, 4H(565, 4H(566, 4H(567, 4H(568, 4H(569, 4H(570, 4H(571, 4H(572,
65 / 4H(573, 4H(574, 4H(575, 4H(576, 4H(577, 4H(578, 4H(579, 4H(580, 4H(581,
66 / 4H(582, 4H(583, 4H(584, 4H(585, 4H(586, 4H(587, 4H(588, 4H(589, 4H(590,
67 / 4H(591, 4H(592, 4H(593, 4H(594, 4H(595, 4H(596, 4H(597, 4H(598, 4H(599,
68 / 4H(600, 4H(601, 4H(602, 4H(603, 4H(604, 4H(605, 4H(606, 4H(607, 4H(608,
69 / 4H(609, 4H(610, 4H(611, 4H(612, 4H(613, 4H(614, 4H(615, 4H(616, 4H(617,
70 / 4H(618, 4H(619, 4H(620, 4H(621, 4H(622, 4H(623, 4H(624, 4H(625, 4H(626,
71 / 4H(627, 4H(628, 4H(629, 4H(630, 4H(631, 4H(632, 4H(633, 4H(634, 4H(635,
72 / 4H(636, 4H(637, 4H(638, 4H(639, 4H(640, 4H(641, 4H(642, 4H(643, 4H(644,
73 / 4H(645, 4H(646, 4H(647, 4H(648, 4H(649, 4H(650, 4H(651, 4H(652, 4H(653,
74 / 4H(654, 4H(655, 4H(656, 4H(657, 4H(658, 4H(659, 4H(660, 4H(661, 4H(662,
75 / 4H(663, 4H(664, 4H(665, 4H(666, 4H(667, 4H(668, 4H(669, 4H(670, 4H(671,
76 / 4H(672, 4H(673, 4H(674, 4H(675, 4H(676, 4H(677, 4H(678, 4H(679, 4H(680,
77 / 4H(681, 4H(682, 4H(683, 4H(684, 4H(685, 4H(686, 4H(687, 4H(688, 4H(689,
78 / 4H(690, 4H(691, 4H(692, 4H(693, 4H(694, 4H(695, 4H(696, 4H(697, 4H(698,
79 / 4H(699, 4H(700, 4H(701, 4H(702, 4H(703, 4H(704, 4H(705, 4H(706, 4H(707,
80 / 4H(708, 4H(709, 4H(710, 4H(711, 4H(712, 4H(713, 4H(714, 4H(715, 4H(716,
81 / 4H(717, 4H(718, 4H(719, 4H(720, 4H(721, 4H(722, 4H(723, 4H(724, 4H(725,
82 / 4H(726, 4H(727, 4H(728, 4H(729, 4H(730, 4H(731, 4H(732, 4H(733, 4H(734,
83 / 4H(735, 4H(736, 4H(737, 4H(738, 4H(739, 4H(740, 4H(741, 4H(742, 4H(743,
84 / 4H(744, 4H(745, 4H(746, 4H(747, 4H(748, 4H(749, 4H(750, 4H(751, 4H(752,
85 / 4H(753, 4H(754, 4H(755, 4H(756, 4H(757, 4H(758, 4H(759, 4H(760, 4H(761,
86 / 4H(762, 4H(763, 4H(764, 4H(765, 4H(766, 4H(767, 4H(768, 4H(769, 4H(770,
87 / 4H(771, 4H(772, 4H(773, 4H(774, 4H(775, 4H(776, 4H(777, 4H(778, 4H(779,
88 / 4H(780, 4H(781, 4H(782, 4H(783, 4H(784, 4H(785, 4H(786, 4H(787, 4H(788,
89 / 4H(789, 4H(790, 4H(791, 4H(792, 4H(793, 4H(794, 4H(795, 4H(796, 4H(797,
90 / 4H(798, 4H(799, 4H(800, 4H(801, 4H(802, 4H(803, 4H(804, 4H(805, 4H(806,
91 / 4H(807, 4H(808, 4H(809, 4H(810, 4H(811, 4H(812, 4H(813, 4H(814, 4H(815,
92 / 4H(816, 4H(817, 4H(818, 4H(819, 4H(820, 4H(821, 4H(822, 4H(823, 4H(824,
93 / 4H(825, 4H(826, 4H(827, 4H(828, 4H(829, 4H(830, 4H(831, 4H(832, 4H(833,
94 / 4H(834, 4H(835, 4H(836, 4H(837, 4H(838, 4H(839, 4H(840, 4H(841, 4H(842,
95 / 4H(843, 4H(844, 4H(845, 4H(846, 4H(847, 4H(848, 4H(849, 4H(850, 4H(851,
96 / 4H(852, 4H(853, 4H(854, 4H(855, 4H(856, 4H(857, 4H(858, 4H(859, 4H(860,
97 / 4H(861, 4H(862, 4H(863, 4H(864, 4H(865, 4H(866, 4H(867, 4H(868, 4H(869,
98 / 4H(870, 4H(871, 4H(872, 4H(873, 4H(874, 4H(875, 4H(876, 4H(877, 4H(878,
99 / 4H(879, 4H(880, 4H(881, 4H(882, 4H(883, 4H(884, 4H(885, 4H(886, 4H(887,
100 / 4H(888, 4H(889, 4H(890, 4H(891, 4H(892, 4H(893, 4H(894, 4H(895, 4H(896,
101 / 4H(897, 4H(898, 4H(899, 4H(900, 4H(901, 4H(902, 4H(903, 4H(904, 4H(905,
102 / 4H(906, 4H(907, 4H(908, 4H(909, 4H(910, 4H(911, 4H(912, 4H(913, 4H(914,
103 / 4H(915, 4H(916, 4H(917, 4H(918, 4H(919, 4H(920, 4H(921, 4H(922, 4H(923,
104 / 4H(924, 4H(925, 4H(926, 4H(927, 4H(928, 4H(929, 4H(930, 4H(931, 4H(932,
105 / 4H(933, 4H(934, 4H(935, 4H(936, 4H(937, 4H(938, 4H(939, 4H(940, 4H(941,
106 / 4H(942, 4H(943, 4H(944, 4H(945, 4H(946, 4H(947, 4H(948, 4H(949, 4H(950,
107 / 4H(951, 4H(952, 4H(953, 4H(954, 4H(955, 4H(956, 4H(957, 4H(958, 4H(959,
108 / 4H(960, 4H(961, 4H(962, 4H(963, 4H(964, 4H(965, 4H(966, 4H(967, 4H(968,
109 / 4H(969, 4H(970, 4H(971, 4H(972, 4H(973, 4H(974, 4H(975, 4H(976, 4H(977,
110 / 4H(978, 4H(979, 4H(980, 4H(981, 4H(982, 4H(983, 4H(984, 4H(985, 4H(986,
111 / 4H(987, 4H(988, 4H(989, 4H(990, 4H(991, 4H(992, 4H(993, 4H(994, 4H(995,
112 / 4H(996, 4H(997, 4H(998, 4H(999, 4H(1000, 4H(1001, 4H(1002, 4H(1003,
113 / 4H(1004, 4H(1005, 4H(1006, 4H(1007, 4H(1008, 4H(1009, 4H(1010, 4H(1011,
114 / 4H(1012, 4H(1013, 4H(1014, 4H(1015, 4H(1016, 4H(1017, 4H(1018, 4H(1019,
115 / 4H(1020, 4H(1021, 4H(1022, 4H(1023, 4H(1024, 4H(1025, 4H(1026, 4H(1027,
116 / 4H(1028, 4H(1029, 4H(1030, 4H(1031, 4H(1032, 4H(1033, 4H(1034, 4H(1035,
117 / 4H(1036, 4H(1037, 4H(1038, 4H(1039, 4H(1040, 4H(1041, 4H(1042, 4H(1043,
118 / 4H(1044, 4H(1045, 4H(1046, 4H(1047, 4H(1048, 4H(1049, 4H(1050, 4H(1051,
119 / 4H(1052, 4H(1053, 4H(1054, 4H(1055, 4H(1056, 4H(1057, 4H(1058, 4H(1059,
120 / 4H(1060, 4H(1061, 4H(1062, 4H(1063, 4H(1064, 4H(1065, 4H(1066, 4H(1067,
121 / 4H(1068, 4H(1069, 4H(1070, 4H(1071, 4H(1072, 4H(1073, 4H(1074, 4H(1075,
122 / 4H(1076, 4H(1077, 4H(1078, 4H(1079, 4H(1080, 4H(1081, 4H(1082, 4H(1083,
123 / 4H(1084, 4H(1085, 4H(1086, 4H(1087, 4H(1088, 4H(1089, 4H(1090, 4H(1091,
124 / 4H(1092, 4H(1093, 4H(1094, 4H(1095, 4H(1096, 4H(1097, 4H(1098, 4H(1099,
125 / 4H(1100, 4H(1101, 4H(1102, 4H(1103, 4H(1104, 4H(1105, 4H(1106, 4H(1107,
126 / 4H(1108, 4H(1109, 4H(1110, 4H(1111, 4H(1112, 4H(1113, 4H(1114, 4H(1115,
127 / 4H(1116, 4H(1117, 4H(1118, 4H(1119, 4H(1120, 4H(1121, 4H(1122, 4H(1123,
128 / 4H(1124, 4H(1125, 4H(1126, 4H(1127, 4H(1128, 4H(1129, 4H(1130, 4H(1131,
129 / 4H(1132, 4H(1133, 4H(1134, 4H(1135, 4H(1136, 4H(1137, 4H(1138, 4H(1139,
130 / 4H(1140, 4H(1141, 4H(1142, 4H(1143, 4H(1144, 4H(1145, 4H(1146, 4H(1147,
131 / 4H(1148, 4H(1149, 4H(1150, 4H(1151, 4H(1152, 4H(1153, 4H(1154, 4H(1155,
132 / 4H(1156, 4H(1157, 4H(1158, 4H(1159, 4H(1160, 4H(1161, 4H(1162, 4H(1163,
133 / 4H(1164, 4H(1165, 4H(1166, 4H(1167, 4H(1168, 4H(1169, 4H(1170, 4H(1171,
134 / 4H(1172, 4H(1173, 4H(1174, 4H(1175, 4H(1176, 4H(1177, 4H(1178, 4H(1179,
135 / 4H(1180, 4H(1181, 4H(1182, 4H(1183, 4H(1184, 4H(1185, 4H(1186, 4H(1187,
136 / 4H(1188, 4H(1189, 4H(1190, 4H(1191, 4H(1192, 4H(1193, 4H(1194, 4H(1195,
137 / 4H(1196, 4H(1197, 4H(1198, 4H(1199, 4H(1200, 4H(1201, 4H(1202, 4H(1203,
138 / 4H(1204, 4H(1205, 4H(1206, 4H(1207, 4H(1208, 4H(1209, 4H(1210, 4H(1211,
139 / 4H(1212, 4H(1213, 4H(1214, 4H(1215, 4H(1216, 4H(1217, 4H(1218, 4H(1219,
140 / 4H(1220, 4H(1221, 4H(1222, 4H(1223, 4H(1224, 4H(1225, 4H(1226, 4H(1227,
141 / 4H(1228, 4H(1229, 4H(1230, 4H(1231, 4H(1232, 4H(1233, 4H(1234, 4H(1235,
142 / 4H(1236, 4H(1237, 4H(1238, 4H(1239, 4H(1240, 4H(1241, 4H(1242, 4H(1243,
143 / 4H(1244, 4H(1245, 4H(1246, 4H(1247, 4H(1248, 4H(1249, 4H(1250, 4H(1251,
144 / 4H(1252, 4H(1253, 4H(1254, 4H(1255, 4H(1256, 4H(1257, 4H(1258, 4H(1259,
145 / 4H(1260, 4H(1261, 4H(1262, 4H(1263, 4H(1264, 4H(1265, 4H(1266, 4H(1267,
146 / 4H(1268, 4H(1269, 4H(1270, 4H(1271, 4H(1272, 4H(1273, 4H(1274, 4H(1275,
147 / 4H(1276, 4H(1277, 4H(1278, 4H(1279, 4H(1280, 4H(1281, 4H(1282, 4H(1283,
148 / 4H(1284, 4H(1285, 4H(1286, 4H(1287, 4H(1288, 4H(1289, 4H(1290, 4H(1291,
149 / 4H(1292, 4H(1293, 4H(1294, 4H(1295, 4H(1296, 4H(1297, 4H(1298, 4H(1299,
150 / 4H(1300, 4H(1301, 4H(1302, 4H(1303, 4H(1304, 4H(1305, 4H(1306, 4H(1307,
151 / 4H(1308, 4H(1309, 4H(1310, 4H(1311, 4H(1312, 4H(1313, 4H(1314, 4H(1315,
152 / 4H(1316, 4H(1317, 4H(1318, 4H(1319, 4H(1320, 4H(1321, 4H(1322, 4H(1323,
153 / 4H(1324, 4H(1325, 4H(1326, 4H(1327, 4H(1328, 4H(1329, 4H(1330, 4H(1331,
154 / 4H(1332, 4H(1333, 4H(1334, 4H(1335, 4H(1336, 4H(1337, 4H(1338, 4H(1339,
155 / 4H(1340, 4H(1341, 4H(1342, 4H(1343, 4H(1344, 4H(1345, 4H(1346, 4H(1347,
156 / 4H(1348, 4H(1349, 4H(1350, 4H(1351, 4H(1352, 4H(1353, 4H(1354, 4H(1355,
157 / 4H(1356, 4H(1357, 4H(1358, 4H(1359, 4H(1360, 4H(1361, 4H(1362, 4H(1363,
158 / 4H(1364, 4H(1365, 4H(1366, 4H(1367, 4H(1368, 4H(1369, 4H(1370, 4H(1371,
159 / 4H(137
```


PL11 106
PL11 107
PL11 108
PL11 109
PL11 110
PL11 111
PL11 112
PL11 113
PL11 114
PL11 115
PL11 116
PL11 117
PL11 118
PL11 119
PL11 120
PL11 121
PL11 122
PL11 123
PL11 124
PL11 125
PL11 126
PL11 127
PL11 128
PL11 129
PL11 130
PL11 131
PL11 132
PL11 133
PL11 134
PL11 135
PL11 136
PL11 137
PL11 138
PL11 139
PL11 140
PL11 141
PL11 142
PL11 143
PL11 144
PL11 145
PL11 146
PL11 147
PL11 148
PL11 149
PL11 150
PL11 151
PL11 152
PL11 153
PL11 154
PL11 155
PL11 156
PL11 157

```

XMI = ARS( XMIN )
XMX = ABS( XMAX )
IF ( XMI .EQ. 0.0 ) XMI = XMX
IF ( XMX .EQ. 0.0 ) XMX = XMI
IF ( XMI .LT. FHL(I) ) GO TO 290
IF ( XMX .GT. FHL(I) ) GO TO 290

```

```

C 270 DO 280 K = 2, 22, 2
      FMT(K) = HH(9)
      FMT(K+1) = HH(1)
      GO TO 310
C 290 DO 300 K = 2, 22, 2
      FMT(K) = HH(10)
      FMT(K+1) = HH(11)

```

```

C 310 YTAG = YMAX - YMIN
      YTAG = YMAX - YMIN
      XTAG = 1*(XMAX-XMIN)
      IF (YTAG.EQ.0.) YTAG=XL
      IF (XTAG.EQ.0.) XTAG=10.
      VFAC=XL/YTAG
      XFACT=10./XTAG
      YMAX1=YMIN*VFAC
      ZMAX1=XMIN*XFAC
      GPT=1.5-ZMAX1
      GM=.5/VFAC
      DO 320 I=1,11

```

```

C 320 S(I)=XMIN+FLOAT(I-1)*XTAG
      IJ=0
      IF ((-ZMAX1).GT.1..AND.(-ZMAX1).LT.99.) IJ=GPT
      IV=1
      LL=50*L+1
      WRITE (5,FMT) ( S (I), I = 1, 11 )
      WRITE (5,L0)
      LAST=25*L
      J=1
      SY = YMAX
      DO 470 LINE=1,LL
      YSAVE = SY
      SY=IFLOAT(LL-LINE)+YMAX1)/VFAC
      S7 = SY + GM

```

```

C 330 DO 330 JI=2,100
      ITRUF(JI)=ME(1)
      IF (IJ.GT.0) ITRUF(IJ)=ME(2)
      ITRUF(1)=ME(2)
      ITRUF(101)=ME(2)
      DO 370 I=1, N
      IF (Y (I) .GE. YSAVE .OR. Y (I) .LT. SY) GO TO 370
      JJ = X (I) * XFACT + GPT
      IF (ITRUF(JJ).EQ.ME(4)) GO TO 370
      JI=3

```

```

      IF (Y(I),LT,SZ) JI = 2
      IF (IRUF(JJ).EQ.ME(1)) GO TO 360
      IF (IRUF(JJ).EQ.ME(2)) GO TO 340
      IF (JI.EQ.3) GO TO 370
      GOT0350
340  IF (JI.EQ.2) GOT0370
350  IRUF(JJ)=ME(4)
      GOT0370
360  IRUF(JJ)=ME(JI)
370  CONTINUE
380  JJ=3
      IF=2
      IF (LINE.NE.1) GO TO 390
      IF (SY.LE.0.) IV = 0
      GO TO 400
390  IF (LINE.EQ.11) GO TO 400
      IF (SY.GT.0..OR.IV.EQ.0) GO TO 430
      IV=0
      IF (SZ.LE.0.) GO TO 400
      JJ=2
      IF=3
      GOT0420 JI=1,101
400  IF (IRUF(JI).EQ.ME(JJ)) GOT0410
      IRUF(JI)=ME(11)
      GOT0420
410  IRUF(JI)=ME(4)
420  CONTINUE
430  IF (KJ1.FQ.0..OR.J.GE.61..OR.(LAST-LINE).GT.30) GO TO 440
      K=KJ(J)
      J=J+1
      IF (L.NE.1) GO TO 450
      IF (J.NE.52) GO TO 450
440  K=ME(1)
450  IF (MCO (LINE,5) .NE.1) GO TO 460
      WRITE (5,FMT) K, SY, (IRUF(1), I = 1, 101), SY
      GOT0470
460  WRITE (5,40) K, (IRUF(1), I = 1, 101)
470  CONTINUE
      WRITE (5,10)
      WRITE (5,FMT) (S(1), I = 1, 11)
      IF (KJ1.NE.0) WRITE (6,50) ( KJ(J), J = 61, 75 )
      RETURN
      PRINT ERROR - THERE WERE AN INVALID NUMBER OF ARGUMENTS
CC
480  WRITE (6,60) NPAP
      RETURN
CC
      PRINT ERROR - XMIN = XMAX OR YMIN = YMAX
490  WRITE (6,70) XMIN, XMAX, YMIN, YMAX
      RETURN

```

```

PL11 158
PL11 159
PL11 160
PL11 161
PL11 162
PL11 163
PL11 164
PL11 165
PL11 166
PL11 167
PL11 168
PL11 169
PL11 170
PL11 171
PL11 172
PL11 173
PL11 174
PL11 175
PL11 176
PL11 177
PL11 178
PL11 179
PL11 180
PL11 181
PL11 182
PL11 183
PL11 184
PL11 185
PL11 186
PL11 187
PL11 188
PL11 189
PL11 190
PL11 191
PL11 192
PL11 193
PL11 194
PL11 195
PL11 196
PL11 197
PL11 198
PL11 199
PL11 200
PL11 201
PL11 202
PL11 203
PL11 204
PL11 205
PL11 206
PL11 207
PL11 208
PL11 209

```

PLYL 210

END


```

54 PRNT
55 PRNT
56 PRNT
57 PRNT
58 PRNT
59 PRNT
60 PRNT
61 PRNT
62 PRNT
63 PRNT
64 PRNT
65 PRNT
66 PRNT
67 PRNT
68 PRNT
69 PRNT
70 PRNT

```

```

JE = JF + NRC
LINES = LINES + 1
IF (IM .GF. M) GO TO 130
IF (LINES .GF. KROW) GO TO 70
GO TO 120
130 IF (JF .GT. JMAX) RETURN
JP = JR3 * MAXR2 + 1
JR2 = JR3 + 1
JR3 = JR3 + R / NRC
IF (KROW .EQ. 47 .AND. NRC .EQ. 1) JR3 = JR3 - 1
IF (JR3 .GF. N) JR3 = N
IM = 0
JE = (JR3 - 1) * MAXR2 + 1
IF (LINES .LT. KROW1) GO TO 110
GO TO 70
140 RETURN
END

```

```

SUBROUTINE RDTAPE (NT,A,IA,NOUT,NER)
  DIMENSION A(IA)
  PFAD (NT), A
  IF (EQF(NT).NE.O) NFR = 51
30 RETURN
END

```

```

KDTAPE 2
RDTAPE 3
PDTAPE 4
RDTAPE 5
RDTAPE 6
RDTAPE 7

```

```

C
SUBROUTINE TRAERO (NOUT,NER,MTOTAP,NTOTAP
1 , CR,D,F,WORK,SPLS,SPLA)
  DIMENSION CR(3,20)
  DIMENSION D(1),WORK(1),F(MTOTAP,1)
  DIMENSION SPLS(MTOTAP,1),SPLA(MTOTAP,1)
  COMMON/DISK2/ND2,ITBL2(843),NRECSA,IBUMP,NKD,V0BWS(20)
  COMMON/XTF/NF(100)
  EQUIVALENCE (NF(6),NSYM), (NF(7),NASYM)
  EQUIVALENCE (NF(41),NK), (NF(42),NG)
  EQUIVALENCE (NF(43),NBOX), (NF(44),NSBETO), (NF(45),NB)
1  DIMENSION GUST(6),AERO(6)
  DATA GUST/4HFPSP,4HFZSP,4HFPAP,4HFZAP,4HFYAP,4HFYAG/
  DATA AERO/4HDPSPG,4HDZSG,4HDYSG,4HDPAG,4HDZAG,4HDYAG/
  NRECSA=2*(NKD*(1+NG)+1)+3
  CALL OPENMS(ND2,ITBL2,NRECSA,0)
  IBUMP=2*(1+NG)
C
  LOC=NKD*IBUMP+3
  CALL RDAERO (4HGEOL,0,WORK,NOJT,NER)
  CALL WRITMS (ND2,WORK,5*NBOX,LOC)
  LOC=LOC+1
  CALL RDAERO (4HGEOS,0,WORK,NOJT,NER)
  CALL WRITMS (ND2,WORK,5*NSBETO,LOC)
C
  CALL RDAERO (4HCR,0,CR,NOUT,NER)
  IF (NER.NE.0) GO TO 1000
  CALL RDAERO (4HRK,0,V0BWS,NOUT,NER)
  IF (NER.NE.0) GO TO 1000
  DO 35 I=1,NKD
    IF (V0BWS(I).EQ.0.0) V0BWS(I) = 1.0E-10
35  V0BWS(I) = 1.0/V0BWS(I)
C
  NOW SAVE SPLS,SPLA
  CALL RDAERO (4HHP,0,D,NOUT,NER)
  N=0
  DO 210 J=1,NSYM
    DO 210 I=1,NBOX
      N=N+1
210  SPLS(I,J) = D(N)
      IF (NASYM.EQ.0) GO TO 220
      CALL RDAERO (4HHPA,0,D,NOUT,NER)
      N=0
      DO 215 J=1,NASYM
        DO 215 I=1,NBOX
          N=N+1
215  SPLA(I,J) = D(N)
220  IF (NB.EQ.0) GO TO 260
      CALL RDAERO (4HHZ,0,D,NOUT,NER)
      I1 = NBOX+1

```

```

TRAERO 2
TRAERO 3
TRAERO 4
TRAERO 5
TRAERO 6
TRAERO 7
DISK2 2
TRAERO 9
TRAERO10
TRAERO11
TRAERO12
TRAERO13
TRAERO14
TRAERO15
TRAERO16
TRAERO17
TRAERO18
TRAERO19
TRAERO20
TRAERO21
TRAERO22
TRAERO23
TRAERO24
TRAERO25
TRAERO26
TRAERO27
TRAERO28
TRAERO29
TRAERO30
TRAERO31
TRAERO32
TRAERO33
TRAERO34
TRAERO35
TRAERO36
TRAERO37
TRAERO38
TRAERO39
TRAERO40
TRAERO41
TRAERO42
TRAERO43
TRAERO44
TRAERO45
TRAERO46
TRAERO47
TRAERO48
TRAERO49
TRAERO50
TRAERO51
TRAERO52
TRAERO53

```

TRAER054
 TRAER055
 TRAER056
 TRAER057
 TRAER058
 TRAER059
 TRAER060
 TRAER061
 TRAER062
 TRAER063
 TRAER064
 TRAER065
 TRAER066
 TRAER067
 TRAER068
 TRAER069
 TRAER070
 TRAER071
 TRAER072
 TRAER073
 TRAER074
 TRAER075
 TRAER076
 TRAER077
 TRAER078
 TRAER079
 TRAER080
 TRAER081
 TRAER082
 TRAER083
 TRAER084
 TRAER085
 TRAER086
 TRAER087
 TRAER088
 TRAER089
 TRAER090
 TRAER091
 TRAER092
 TRAER093
 TRAER094
 TRAER095
 TRAER096
 TRAER097
 TRAER098
 TRAER099
 TRAER100
 TRAER101
 TRAER102
 TRAER103
 TRAER104
 TRAER105

```

    I2 = NBOX+NSBETO
    N = 0
    DO 230 J=1,NSYM
      DO 230 I=1,I2
        N = N+1
        230 SPLS(I,J) = D(N)
        IF (NASYM.EQ.0) GO TO 240
        CALL RDAERO (4HHZA ,0,D,NOUT,NER)
        N = 0
        DO 235 J=1,NASYM
          DO 235 I=1,I2
            N = N+1
            235 SPLA(I,J) = D(N)
            240 CALL RDAERO (4HHYS ,0,D,NOUT,NER)
            I1 = I2+1
            I2 = I1+NSBETO-1
            N = 0
            DO 250 J=1,NSYM
              DO 250 I=1,I2
                N = N+1
                250 SPLS(I,J) = D(N)
                IF (NASYM.EQ.0) GO TO 260
                CALL RDAERO (4HHYA ,0,D,NOUT,NER)
                N = 0
                DO 255 J=1,NASYM
                  DO 255 I=1,I2
                    N = N+1
                    255 SPLA(I,J) = D(N)
                    C
                    260 NW=NTOTAP*NSYM
                    LOC=NKD*IBUMP+1
                    CALL WRITMS(ND2,SPLS,NW,LOC)
                    NW=NTOTAP*NASYM
                    LOC=LOC+1
                    CALL WRITMS(ND2,SPLA,NW,LOC)
                    C
                    DO 200 K=1,NKD
                      C
                      KBUMP=(K-1)*IBUMP
                      LOCO=KBUMP
                      M=2
                      IG = 0
                      C
                      110 NAERO = 2*NBOX
                      I2=0
                      M=M+1
                      DO 140 MB=1,3
                        IG = IG+1
                        I1 = I2+1
                        I2 = I1+NAERO-1
                        IF (MB.GT.1.AND.NR.EQ.0) GO TO 140
                        C
                        READ GUST DATA
  
```

```

C      CALL RDAERO (GUST(IG),K,WORK,NOUT,NER)
      N = 0
      DO 130 J=1,NG
      DO 130 I=1,I2
      N = N+1
      130 F(I,J) = WORK(N)
C
      NAERO = 2*NSBETO
      140 CONTINUE
C
      DO 135 J=1,NG
      LOC=LOC0+M+2*(J-1)
      135 CALL WRITMS(ND2,F(1,J),MTOTAP,LOC)
C
      IF (NASYM.EQ.0) GO TO 150
      IF (M.GT.3) GO TO 150
      GO TO 110
C
      GENERALIZED AERO
C
      150 NM = NSYM
      IND = 1
      M=1
      IA = 1
C
      160 NWD = 2*NM*NM
      CALL RDAERO (AERO(IA),K,D,NOUT,NER)
      IF (NB.EQ.0) GO TO 100
      DO 90 N=1,2
      IA = IA+1
      CALL RDAERO (AERO(IA),K,WORK,NOUT,NER)
      DO 75 I=1,NWD
      75 D(I) = D(I) + WORK(I)
      90 CONTINUE
C
      100 LOC=M+KBUMP
      CALL WRITMS(ND2,D,NWD,LOC)
C
      IF (NASYM.EQ.0) GO TO 200
      IF (IND.EQ.3) GO TO 200
      NM = NASYM
      IND = 3
      M=2
      IA = 4
      GO TO 160
C
      200 CONTINUE
C
      1000 RETURN
      END

```

```

TRAER106
TRAER107
TRAER108
TRAER109
TRAER110
TRAER111
TRAER112
TRAER113
TRAER114
TRAER115
TRAER116
TRAER117
TRAER118
TRAER119
TRAER120
TRAER121
TRAER122
TRAER123
TRAER124
TRAER125
TRAER126
TRAER127
TRAER128
TRAER129
TRAER130
TRAER131
TRAER132
TRAER133
TRAER134
TRAER135
TRAER136
TRAER137
TRAER138
TRAER139
TRAER140
TRAER141
TRAER142
TRAER143
TRAER144
TRAER145
TRAER146
TRAER147
TRAER148
TRAER149
TRAER150
TRAER151
TRAER152
TRAER153
TRAER154
TRAER155
TRAER156

```

```

SUBROUTINE WRAERO (ITYP,IK,IHD,NDUT,NER)
DIMENSION IHD(50)

COMMON /DOTRLS/DOTRL(20,10)
INTEGER DOTRL
LOGICAL READ

DIMENSION TYPE(31)
DATA TYPE/4HHEAD,4HGEOM,4HGE01,4HGE03,4HGE05,4HCR,4HRK
1,4HHPS,4HHPA,4HHZS,4HHZA,4HHYS,4HHYA
2,4HDPSP,4HFPSP,4HDZSP,4HFZSP,4HDYSP,4HFYSP
3,4HDPAP,4HFPAP,4HDZAP,4HFZAP,4HDYAP,4HFYAP
4,4HDPSC,4HDZSC,4HDYSC,4HDPAG,4HDZAG,4HDYAG/
DATA ITYP/4HAERO/

READ = .FALSE.
GO TO 10

ENTRY RDAERO
READ = .TRUE.

10 NER = 0
IF (J.NE.0) GO TO 20
CALL FDOTRL(ITYP,J,J0)
IF (J.NE.0) GO TO 20
NFP = 54
GO TO 1000

20 NT = DOTRL(I,J)
IF (NT.EQ.0) GO TO 1000
IF (DOTRL(20,J).EQ.0) GO TO 50
WRITE (NDUT,1) ITYP
1 FORMAT (IHO,A4,* FILE ERROR CONDITION - FILE CANNOT BE PROCESSED*)
GO TO 1000

50 DO 60 I=1,31
ITYP = I
IF (ITYP.EQ.TYPE(I)) GO TO 90
60 CONTINUE
GO TO 900

90 NSYM = DOTRL(11,J)
NS = 2
IF (NASYM.EQ.0) NS=1
NASYM = DOTRL(12,J)
NGUST = DOTRL(13,J)
NAGUST = NGUST
IF (NASYM.EQ.0) NAGUST=0
NB = 3
IF (DOTRL(14,J).EQ.0) NB=1
IF (NB.EQ.3) GO TO 95

```

WRAERO 2
WRAERO 3
WRAERO 4
DDTBL 2
DDTBL 3
WRAERO 6
WRAERO 7
WRAERO 8
WRAERO 9
WRAERO 10
WRAERO 11
WRAERO 12
WRAERO 13
WRAERO 14
WRAERO 15
WRAERO 16
WRAERO 17
WRAERO 18
WRAERO 19
WRAERO 20
WRAERO 21
WRAERO 22
WRAERO 23
WRAERO 24
WRAERO 25
WRAERO 26
WRAERO 27
WRAERO 28
WRAERO 29
WRAERO 30
WRAERO 31
WRAERO 32
WRAERO 33
WRAERO 34
WRAERO 35
WRAERO 36
WRAERO 37
WRAERO 38
WRAERO 39
WRAERO 40
WRAERO 41
WRAERO 42
WRAERO 43
WRAERO 44
WRAERO 45
WRAERO 46
WRAERO 47
WRAERO 48
WRAERO 49
WRAERO 50
WRAERO 51
WRAERO 52

WRAER053
 WRAER054
 WRAER055
 WRAER056
 WRAER057
 WRAER058
 WRAER059
 WRAER060
 WRAER061
 WRAER062
 WRAER063
 WRAER064
 WRAER065
 WRAER066
 WRAER067
 WRAER068
 WRAER069
 WRAER070
 WRAER071
 WRAER072
 WRAER073
 WRAER074
 WRAER075
 WRAER076
 WRAER077
 WRAER078
 WRAER079
 WRAER080
 WRAER081
 WRAER082
 WRAER083
 WRAER084
 WRAER085
 WRAER086
 WRAER087
 WRAER088
 WRAER089
 WRAER090
 WRAER091
 WRAER092
 WRAER093
 WRAER094
 WRAER095
 WRAER096
 WRAER097
 WRAER098
 WRAER099
 WRAER100
 WRAER101
 WRAER102
 WRAER103
 WRAER104

```

    IF (ITYPR.EQ.5) GO TO 900
    IF (ITYPR.GT.9.AND.ITYPR.LT.14) GO TO 900
    IF (ITYPR.GT.15.AND.ITYPR.LT.20) GO TO 900
    IF (ITYPR.GT.21.AND.ITYPR.LT.26) GO TO 900
    IF (ITYPR.EQ.27.CR.ITYPR.EQ.28) GO TO 900
    IF (ITYPR.EQ.30.OR.ITYPR.EQ.31) GO TO 900

C   95 IF (NASYM.NE.0) GO TO 100
    IF (ITYPR.EQ.9.OR.ITYPR.EQ.11.OR.ITYPR.EQ.13) GO TO 900
    IF (ITYPR.GT.19.AND.ITYPR.LT.26) GO TO 900
    IF (ITYPR.GT.28.AND.ITYPR.LT.32) GO TO 900

C   100 IF (ITYPR.GT.7) GO TO 150

C   IRN = ITYPR
    NREAD = 1
    GO TO (101,102,103,104,105,106,107), ITYPR

C   101 NW = 50
    GO TO 200
    102 NW = DDTBL(15,J)*3 + DDTBL(17,J)*8 + DDTBL(18,J)
    GO TO 200
    103 NW = DDTBL(15,J)*6
    GO TO 200
    104 NW = DDTBL(15,J)*3
    GO TO 200
    105 NW = DDTBL(16,J)*6
    GO TO 200
    106 NW = 3*DDTBL(13,J)
    IF (NB.EQ.1) IRN = IRN-1
    GO TO 200
    107 NW = DDTBL(10,J)
    IF (NB.EQ.1) IRN = IRN-1
    GO TO 200

C   150 IRN = 7
    IF (NB.EQ.3) IRN = IRN+1
    IF (ITYPR.GT.13) GO TO 180
    INTERPOLATION MATRICES

C   NW = DDTBL(15,J)
    AREAD = NSYM
    IF (ITYPR.EQ.8) GO TO 200
    IRN = IRN+NSYM
    AREAD = NASYM
    IF (ITYPR.EQ.9) GO TO 200
    NW = DDTBL(15,J)
    IRN = IRN+NASYM
    AREAD = NSYM
    IF (ITYPR.EQ.10) GO TO 200
    IRN = IRN+NSYM
  
```

```

      NREAD = NASYM
      IF (ITYPR.EQ.11) GO TO 200
      IRN = IRN+NASYM
      NREAD = NSYM
      IF (ITYPR.EQ.12) GO TO 200
      IRN = IRN+NSYM
      NREAD = NASYM
      GO TO 200

C 180 IPN = IRN + (NSYM+NASYM)*NR
      IRN = IRN + (IK-1)*NB*(NSYM+NGUST+NASYM+VAGUST+NS)
      IF (ITYPR.GT.25) GO TO 190
      PHYSICAL FORCES

      NW = DDTRL(15,J)*2
      NREAD = NSYM
      IF (ITYPR.EQ.14) GO TO 200
      IRN = IRN+NSYM
      NREAD = NGUST
      IF (ITYPR.EQ.15) GO TO 200
      NW = DDTRL(16,J)*2
      IRN = IRN+NGUST
      NREAD = NSYM
      IF (ITYPR.EQ.16) GO TO 200
      IRN = IRN+NSYM
      NREAD = NGUST
      IF (ITYPR.EQ.17) GO TO 200
      IRN = IRN+NGUST
      NREAD = NSYM
      IF (ITYPR.EQ.18) GO TO 200
      IRN = IRN+NSYM
      NREAD = NGUST
      IF (ITYPR.EQ.19) GO TO 200
      NW = DDTRL(15,J)*2
      IRN = IRN+NGUST
      NREAD = NASYM
      IF (ITYPR.EQ.20) GO TO 200
      IRN = IRN+NASYM
      NREAD = NGUST
      IF (ITYPR.EQ.21) GO TO 200
      NW = DDTRL(16,J)*2
      IRN = IRN+NGUST
      NREAD = NASYM
      IF (ITYPR.EQ.22) GO TO 200
      IRN = IRN+NSYM
      NREAD = NGUST
      IF (ITYPR.EQ.23) GO TO 200
      IRN = IRN+NGUST
      NREAD = NASYM
      IF (ITYPR.EQ.24) GO TO 200
      IRN = IRN+NASYM
      NREAD = NGUST

```

WRAERI05
 WRAERI06
 WRAERI07
 WRAERI08
 WRAERI09
 WRAERI10
 WRAERI11
 WRAERI12
 WRAERI13
 WRAERI14
 WRAERI15
 WRAERI16
 WRAERI17
 WRAERI18
 WRAERI19
 WRAERI20
 WRAERI21
 WRAERI22
 WRAERI23
 WRAERI24
 WRAERI25
 WRAERI26
 WRAERI27
 WRAERI28
 WRAERI29
 WRAERI30
 WRAERI31
 WRAERI32
 WRAERI33
 WRAERI34
 WRAERI35
 WRAERI36
 WRAERI37
 WRAERI38
 WRAERI39
 WRAERI40
 WRAERI41
 WRAERI42
 WRAERI43
 WRAERI44
 WRAERI45
 WRAERI46
 WRAERI47
 WRAERI48
 WRAERI49
 WRAERI50
 WRAERI51
 WRAERI52
 WRAERI53
 WRAERI54
 WRAERI55
 WRAERI56

```

C      GO TO 200
C 190 IRN = IRN + NR*(NSYM+NGUST+NASYM+NAGUST)
C      GENERALIZED FORCES
C      IRN = IRN+ITYPR-26
C      NFEAD = 1
C      NW = NSYM*(NSYM+NGUST)
C      IF (ITYPR.LT.29) GO TO 200
C      NW = NASYM*(NASYM+NAGUST)
C      GO TO 200
C 200 IF (READ) GO TO 250
C      IF (DDTBL(4,J).EQ.0) GO TO 210
C      NER = 1
C      GO TO 1000
C 210 IRN = DDTBL(5,J) + 1
C      IF (ITYPR.EQ.1) IHD(1) = ITYP
C      CALL WRTAPE (NT,IHD,NW,NOUT,NER)
C      IF (NER.NE.0) GO TO 990
C      GO TO 500
C 250 DDTBL(4,J) = 1
C      LPN = DDTBL(5,J)
C      NSKIP = IRN-LRN-1
C      IF (NSKIP.EQ.0) GO TO 300
C      IF (NSKIP.GT.0) GO TO 280
C      DEWIND NT
C      WRITE (NCUT,9) NT
C      9 FORMAT (1H0,5(4H****)/5H0TAPE,12,8H REWOUND/1H0,5(4H****) )
C      IF (ITYPR.EQ.1) GO TO 300
C      NSKIP = IRN-1
C 280 DO 290 I=1,NSKIP
C 290 READ(NT)
C 300 IPN = IRN-1
C      N = 1
C      DO 310 I=1,NREAD
C      CALL ROTAPE (NT,IHD(N),NW,NOUT,NER)
C      IF (NER.NE.0) GO TO 990
C      IRN = IRN+1
C      N = N+NW
C 310
C 500 DDTBL(5,J) = IRN
C      IF (ITYPR.NE.1) GO TO 1000
C      IF (IHD(1).EQ.ITYPR) GO TO 510
C      NER = 1
C      WRITE (NCUT,501) ITYP,IHD(1)
C      501 FORMAT (1H0,*INVALID FILE#/1H0,*FILE SHOULD BE =*,A4

```

WRAER157
 WRAER158
 WRAER159
 WRAER160
 WRAER161
 WRAER162
 WRAER163
 WRAER164
 WRAER165
 WRAER166
 WRAER167
 WRAER168
 WRAER169
 WRAER170
 WRAER171
 WRAER172
 WRAER173
 WRAER174
 WRAER175
 WRAER176
 WRAER177
 WRAER178
 WRAER179
 WRAER180
 WRAER181
 WRAER182
 WRAER183
 WRAER184
 WRAER185
 WRAER186
 WRAER187
 WRAER188
 WRAER189
 WRAER190
 WRAER191
 WRAER192
 WRAER193
 WRAER194
 WRAER195
 WRAER196
 WRAER197
 WRAER198
 WRAER199
 WRAER200
 WRAER201
 WRAER202
 WRAER203
 WRAER204
 WRAER205
 WRAER206
 WRAER207
 WRAER208

WRAER209
WRAER210
WRAER211
WRAER212
WRAER213
WRAER214
WRAER215
WRAER216
WRAER217
WRAER218
WRAER219
WRAER220
WRAER221
WRAER222
WRAER223
WRAER224
WRAER225
WRAER226
WRAER227
WRAER228
WRAER229
WRAER230
WRAER231
WRAER232
WRAER233

```

1 /14,*FILE IS      =*,A4)
CC TO 990

510 DO 520 I=2,11
520 DCTRL(I+8,J) = IHD(I)

      IF (.NOT.READ) WRITE (NCUT,3) NT
3  FORMAT (28H HEADER DATA WRITTEN ON TAPE,I2)
      IF (.READ) WRITE (NCUT,4) NT
4  FORMAT (28H HEADER DATA READ FROM TAPE,I2)
      WRITE (NCUT,5) (IHD(I),I=2,11), (IHD(I),I=21,25)
5  FORMAT
1  (8X2H NK,5X4H NSYM,5X5H NASYM,8X2H NR,6X4H NRQX,6X4H NSHE
2  , 4X6H NSTRIP,4X6H MAXSTR,8X2H NP /10I10/
3  / 6X4H MACH,6X4H REFFA,6X4H REFFS,6X4H REFFC,8X2H X4 /5F10.2 )
GC TO 1000

C 900 NER = 1
      WRITE (NCUT,2) TYPR
2  FORMAT (IHO,*INVALID REQUEST FROM AERO FILE FOR RECORD TYPE=*,A4)

C 990 DCTRL(20,J) = 1
C 1000 RETURN
      END

```

```

SUBROUTINE WRFRSP (IR,IG,IK,IHD,NCUT,NFR)
DIMENSION IHD(50)
COMMON /DDTBLS/DDTBL(20,10)
INTEGER DDTBL
LOGICAL READ
DATA ITP/4HFRSP/

C
  READ = .FALSE.
  GO TO 10
  ENTRY RDRSP
  READ = .TRUE.

C
  10 IF (J.NE.0) GO TO 20
  CALL FDDTBL(ITP,JI,J)
  IF (J.NE.0) GO TO 20
  NFR = 1
  GO TO 1000

C
  20 NT = DDTBL(1,J)
  IF (NT.EQ.0) GO TO 1000
  IF (DDTBL(20,J).EQ.0) GO TO 50
  WRITE (NOUT,1) ITP
  1 FFORMAT (140,A4, FILE ERRCR CONDITION - FILE CANNOT BE PROCESSED )
  NFR = 1
  GO TO 1000

C
  50 IF (IR.GT.0.AND.IR.LT.10) GO TO 100
  NFR = 1
  GO TO 1000

C
  100 IF (IR.EQ.1) NW=50
  IF (IR.EQ.2) NW = DDTBL(11,J)*3
  IF (IR.EQ.3) NW = DDTBL(12,J)
  IF (IR.EQ.4) NW=5
  IF (IR.EQ.5) NW = DDTBL(15,J)
  IF (IR.EQ.6) NW = DDTBL(15,J)
  IF (IR.EQ.7) NW = 2*(DDTBL(13,J)+DDTBL(14,J) )
  IF (IR.EQ.8) NW = 2*DDTBL(16,J)
  IF (IR.EQ.9) NW = 2*DDTBL(16,J)

C
  IF (READ) GO TO 200
  IF (DDTBL(4,J).EQ.0) GO TO 110
  NFR = 1
  GO TO 1000

C
  110 IRN = DDTBL(5,J)+1
  IF (IR.EQ.1) IHD(1) = ITP

C
  150 CALL WRTAPE (NT,IHD,NW,NCUT,NFR)
  GO TO 500

C
  200 IRN = IR

```

```

WRFRSP 2
WRFRSP 3
DDTBLS 2
DDTBLS 3
WRFRSP 5
WRFRSP 6
WRFRSP 7
WRFRSP 8
WRFRSP 9
WRFRSP10
WRFRSP11
WRFRSP12
WRFRSP13
WRFRSP14
WRFRSP15
WRFRSP16
WRFRSP17
WRFRSP18
WRFRSP19
WRFRSP20
WRFRSP21
WRFRSP22
WRFRSP23
WRFRSP24
WRFRSP25
WRFRSP26
WRFRSP27
WRFRSP28
WRFRSP29
WRFRSP30
WRFRSP31
WRFRSP32
WRFRSP33
WRFRSP34
WRFRSP35
WRFRSP36
WRFRSP37
WRFRSP38
WRFRSP39
WRFRSP40
WRFRSP41
WRFRSP42
WRFRSP43
WRFRSP44
WRFRSP45
WRFRSP46
WRFRSP47
WRFRSP48
WRFRSP49
WRFRSP50
WRFRSP51
WRFRSP52

```

WRER SP53
 WRER SP54
 WRER SP55
 WRER SP56
 WRER SP57
 WRER SP58
 WRER SP59
 WRER SP60
 WRER SP61
 WRER SP62
 WRER SP63
 WRER SP64
 WRER SP65
 WRER SP66
 WRER SP67
 WRER SP68
 WRER SP69
 WRER SP70
 WRER SP71
 WRER SP72
 WRER SP73
 WRER SP74
 WRER SP75
 WRER SP76
 WRER SP77
 WRER SP78
 WRER SP79
 WRER SP80
 WRER SP81
 WRER SP82
 WRER SP83
 WRER SP84
 WRER SP85
 WRER SP86
 WRER SP87
 WRER SP88
 WRER SP89
 WRER SP90
 WRER SP91
 WRER SP92
 WRER SP93
 WRER SP94
 WRER SP95
 WRER SP96
 WRER SP97
 WRER SP98

```

IF (IR,LT,4) GO TO 250
IFN = IP. + 6*(IK-I + DDIBL(12,J))*(IG-1) )

C 250 LPN = DDIBL(5,J)
      NRSKP = IRN-LFN-1
      IF (NRSKP.EQ.0) GO TO 300
      IF (NRSKP.GT.0) GO TO 280
      REWIND NT
      WRITE (NOUT,2) NT
      2 FORMAT (1H,5(4H****)/5HOTAPE,12,8H REWOUND/1H0,5(4H****) )
      IF (IR.FQ.1) GO TO 300
      NRSKP = IRN-1

C 280 DO 290 I=1,NRSKP
      290 READ(NT)

C 300 CALL RDTAPE (NT,IHD,NW,NOUT,NER)
      IF (NER.NE.0) GO TO 1000
      DDIBL(4,J) = 1

C 500 IF (IR.GT.1) GO TO 550
      IF (IHD(1).EQ.ITYP) GO TO 510
      NER = 1
      WRITE (NOUT,501) ITYP, IHD(1)
      501 FORMAT (1H, INVALID FILE /1H0, FILE SHOULD BE = ,A4)
      1 /1H, FILE IS
      GO TO 990

C 510 DO 520 I=3,8
      DDIBL(I+8,J) = IHD(I)
      520 IF (.NOT.PEAD) WRITE (NOUT,3) NT
      IF (.PEAD) WRITE (NOUT,4) NT
      WRITE (NOUT,5) (IHD(I),I=2,7), (IHD(1),I=11,12)
      3 FORMAT (28H HEADER DATA WRITTEN ON TAPE,12)
      4 FORMAT (28H HEADER DATA READ FROM TAPE,12)
      5 FORMAT (5X5HIDENT,5X5HNGUST,5X5HNFRFQ,6X4HNSYM,5X5HNASYM,7X3HNKI
      2 , 7X3HVEL,5X,5HSIGMA
      3 / 6110,2F10.2)

C 550 DDIBL(5,J) = IRN
      GO TO 1000

C 990 DDIBL(20,J) = 1
      1000 RETURN
      END
  
```

```

SUBROUTINE WRLOAD (IR,IG,IHD,NOUT,NER)
DIMENSION IHD(50)
COMMON /DDTBL/ DDTBL(20,10)
INTEGER DDTBL
LOGICAL READ
DATA ITYP/4HLOAD/

C
READ = .FALSE.
GO TO 10
ENTRY RDLOAD
READ = .TRUE.

C
10 IF (J.NE.0) GO TO 20
CALL FDDTBL(ITYP,JI,J)
IF (J.NE.0) GO TO 20
NER = 1
GO TO 1000

C
20 NT = DDTBL(1,J)
IF (NT.EQ.0) GO TO 1000
IF (DDTBL(20,J).EQ.0) GO TO 50
WRITE (NOUT,1) ITYP
1 PCRMAT (140,A4, FILE ERROR CONDITION - FILE CANNOT BE PROCESSED
NER = 1
GO TO 1000

C
50 IF (IR.ST.O.AND.IR.LT.10) GO TO 100
NER = 1
GO TO 1000

C
100 CONTINUE
IF (IR.EQ.1) NW=50
IF (IR.EQ.2) NW = DDTBL(13,J)*8
IF (IR.EQ.3) NW = 3*DDTBL(11,J)
IF (IR.EQ.4) NW = DDTBL(12,J)
IF (IR.EQ.5) NW = 2*DDTBL(14,J)
IF (IR.EQ.6) NW = 2*DDTBL(14,J)*DDTBL(12,J)
IF (IR.EQ.7) NW = 2*DDTBL(14,J)*DDTBL(12,J)
IF (IR.EQ.8) NW = 2*DDTBL(13,J)*DDTBL(12,J)
IF (IR.EQ.9) NW = 2*DDTBL(13,J)*DDTBL(12,J)

C
IF (READ) GO TO 200
IF (DDTBL(4,J).EQ.0) GO TO 110
NER = 1
GO TO 1000

C
110 IRN = DDTBL(5,J)+1
IF (IR.EQ.1) IHD(1) = ITYP
CALL WRTAPE (NT,IHD,NW,NOUT,NFR)
GO TO 500

C
200 IRN = IR

```

```

WRLOAD 2
WRLOAD 3
DDTBL 2
DDTBL 3
WRLOAD 5
WRLOAD 6
WRLOAD 7
WRLOAD 8
WRLOAD 9
WRLOAD 10
WRLOAD 11
WRLOAD 12
WRLOAD 13
WRLOAD 14
WRLOAD 15
WRLOAD 16
WRLOAD 17
WRLOAD 18
WRLOAD 19
WRLOAD 20
WRLOAD 21
WRLOAD 22
WRLOAD 23
WRLOAD 24
WRLOAD 25
WRLOAD 26
WRLOAD 27
WRLOAD 28
WRLOAD 29
WRLOAD 30
WRLOAD 31
WRLOAD 32
WRLOAD 33
WRLOAD 34
WRLOAD 35
WRLOAD 36
WRLOAD 37
WRLOAD 38
WRLOAD 39
WRLOAD 40
WRLOAD 41
WRLOAD 42
WRLOAD 43
WRLOAD 44
WRLOAD 45
WRLOAD 46
WRLOAD 47
WRLOAD 48
WRLOAD 49
WRLOAD 50
WRLOAD 51
WRLOAD 52

```

```

IF (IR.LT.5) GO TO 250
M = 4
IF (DDTBL(14,J).NE.0) GO TO 220
IRN = IRN-3
M = 2
IF (IR.EQ.8.OR.IR.EQ.9) GO TO 220
NER = 201
WRITE (NOUT,6) TYPR
6 FORMAT (I10,*INVALID REQUEST FROM AERC FILE FOR RECORD TYPE=*,A4)
GO TO 990
C
C 220 IRN = IRN + M*(IG-1)
C
C 250 LRN = DDTBL(5,J)
NRSKP = IRN-LRN-1
IF (NRSKP.EQ.0) GO TO 300
IF (NRSKP.GT.0) GO TO 280
REWIND NT
WRITE (NOUT,2) NT
2 FORMAT (I10,5(4H***)/5HOTAPE,I2,8H REWOUND/I10,5(4H***) )
IF (IR.FQ.1) GO TO 300
NRSKP = IRN-1
C
C 280 DO 290 I=1,NRSKP
C 290 READ(NT)
C
C 300 CALL ROTAPE (NT,IHD,NW,NOUT,NER)
IF (NER.NE.0) GO TO 1000
DDTBL(4,J) = 1
C
C 500 IF (IR.GT.1) GO TO 550
IF (IHD(1).EQ.ITYP) GO TO 510
NER = 1
WRITE (NOUT,501) ITYP, IHD(1)
501 FORMAT (I10,'INVALID FILE/I10, FILE SHOULD BE = ,A4)
1 /14, FILE IS = ,A4)
GO TO 990
C
C 510 DO 520 I=3,6
C 520 DDTBL(1+8,J) = IHD(I)
IF (.NOT.READ) WRITE (NOUT,3) NT
IF (.PEAD) WRITE (NOUT,4) NT
WRITE (NOUT,5) (IHD(I),I=2,6), (IHD(I),I=11,12)
3 FORMAT (28H HEADER DATA WRITTEN ON TAPE,I2)
4 FORMAT (28H HEADER DATA READ FROM TAPE,I2)
5 FORMAT (5X5HIDENT,5X5HNGUST,5X5HNFREQ,4X6HINTLD,6X4HNACC
1 ,7X3HVEL,5X5HSIGMA
2 /5I10,2F10.2)
C
C 550 DDTBL(5,J) = IRN
GO TO 1000
C

```

```

WRL0AD53
WRL0AD54
WRL0AD55
WRL0AD56
WRL0AD57
WRL0AD58
WRL0AD59
WRL0AD60
WRL0AD61
WRL0AD62
WRL0AD63
WRL0AD64
WRL0AD65
WRL0AD66
WRL0AD67
WRL0AD68
WRL0AD69
WRL0AD70
WRL0AD71
WRL0AD72
WRL0AD73
WRL0AD74
WRL0AD75
WRL0AD76
WRL0AD77
WRL0AD78
WRL0AD79
WRL0AD80
WRL0AD81
WRL0AD82
WRL0AD83
WRL0AD84
WRL0AD85
WRL0AD86
WRL0AD87
WRL0AD88
WRL0AD89
WRL0AD90
WRL0AD91
WRL0AD92
WRL0AD93
WRL0AD94
WRL0AD95
WRL0AD96
WRL0AD97
WRL0AD98
WRL0AD99
WRL0A100
WRL0A101
WRL0A102
WRL0A103
WRL0A104

```

WRLOA105
WRLOA106
WRLOA107
WRLOA108

990 DD TBL (20,J) = 1
C 1000 RETURN
END

SUBROUTINE WRTAPE (NT,A,IA,NOUT,NER)
DIMENSION A(IA)
WRITE (NT) A
RETURN
END

WRTAPE 2
WRTAPE 3
WRTAPE 4
WRTAPE 5
WRTAPE 6

```

SUBROUTINE WRUNIT (IR,IK,IHD,NOUT,NER)
DIMENSION IHD(50)
COMMON /DDTBLS/DDTBL(20,10)
INTEGER DDTBL
LOGICAL READ
DATA ITYP/4HUNIT/

READ = .FALSE.
GO TO 10
ENTRY RUNIT
READ = .TRUE.

10 IF (J.NE.0) GO TO 20
CALL FDDTBL(ITYP,JI,J)
IF (J.NE.0) GO TO 20
NEP = 1
GO TO 1000

20 NT = DDTBL(1,J)
IF (NT.EQ.0) GO TO 1000
IF (DDTBL(20,J).EQ.0) GO TO 50
WRITE (NOUT,1) ITYP
1 FFORMAT (140,A4, FILE ERROR CONDITION - FILE CANNOT BE PROCESSED )
NEP = 1
GO TO 1000

50 IF (IR.GT.0.AND.IR.LT.16) GO TO 100
NEP = 1
GO TO 1000

100 CONTINUE
IF (IR.EQ.1) NW=50
IF (IR.EQ.2) NW = DDTBL(10,J)*8
IF (IR.EQ.3) NW = DDTBL(11,J)*DDTBL(10,J)
IF (IR.EQ.4) NW = DDTBL(10,J)*DDTBL(12,J)
IF (IR.EQ.5) NW = DDTBL(13,J)*DDTBL(12,J)
IF (IR.EQ.6) NW = DDTBL(10,J)*DDTBL(13,J)+DDTBL(14,J)
IF (IR.EQ.7) NW = 2*DDTBL(10,J)*DDTBL(13,J)
IF (IR.EQ.8) NW = 2*DDTBL(10,J)*DDTBL(14,J)
IF (IR.EQ.9) NW = DDTBL(10,J)
IF (IR.EQ.10) NW = DDTBL(10,J)
IF (IR.EQ.11) NW = 3*DDTBL(16,J)
IF (IR.EQ.12) NW = DDTBL(10,J)*DDTBL(16,J)+DDTBL(18,J)
IF (IR.EQ.13) NW = 3*DDTBL(17,J)
IF (IR.EQ.14) NW = DDTBL(10,J)*DDTBL(17,J)+DDTBL(19,J)

IF (READ) GO TO 200
IF (DDTBL(4,J).EQ.0) GO TO 110
NEP = 1
GO TO 990

110 IFN = DDTBL(5,J)+1

```

WRUNIT 2
 WRUNIT 2
 DDTBLS 3
 DDTBLS 3
 WRUNIT 5
 WRUNIT 6
 WRUNIT 7
 WRUNIT 8
 WRUNIT 9
 WRUNIT 10
 WRUNIT 11
 WRUNIT 11
 WRUNIT 12
 WRUNIT 13
 WRUNIT 14
 WRUNIT 15
 WRUNIT 16
 WRUNIT 17
 WRUNIT 18
 WRUNIT 19
 WRUNIT 20
 WRUNIT 21
 WRUNIT 22
 WRUNIT 23
 WRUNIT 24
 WRUNIT 25
 WRUNIT 26
 WRUNIT 27
 WRUNIT 28
 WRUNIT 29
 WRUNIT 30
 WRUNIT 31
 WRUNIT 32
 WRUNIT 33
 WRUNIT 34
 WRUNIT 35
 WRUNIT 36
 WRUNIT 37
 WRUNIT 38
 WRUNIT 39
 WRUNIT 40
 WRUNIT 41
 WRUNIT 42
 WRUNIT 43
 WRUNIT 44
 WRUNIT 45
 WRUNIT 46
 WRUNIT 47
 WRUNIT 48
 WRUNIT 49
 WRUNIT 50
 WRUNIT 51
 WRUNIT 52

WRUNI T53
WRUNI T54
WRUNI T55
WRUNI T56
WRUNI T57
WRUNI T58
WRUNI T59
WRUNI T60
WRUNI T61
WRUNI T62
WRUNI T63
WRUNI T64
WRUNI T65
WRUNI T66
WRUNI T67
WRUNI T68
WRUNI T69
WRUNI T70
WRUNI T71
WRUNI T72
WRUNI T73
WRUNI T74
WRUNI T75
WRUNI T76
WRUNI T77
WRUNI T78
WRUNI T79
WRUNI T80
WRUNI T81
WRUNI T82
WRUNI T83
WRUNI T84
WRUNI T85
WRUNI T86
WRUNI T87
WRUNI T88
WRUNI T89
WRUNI T90
WRUNI T91
WRUNI T92
WRUNI T93
WRUNI T94
WRUNI T95
WRUNI T96
WRUNI T97
WRUNI T98
WRUNI T99
WRUNI T100
WRUNI T101
WRUNI T102
WRUNI T103
WRUNI T104

```

IF (IR.EQ.1) IHD(1) = ITYP
CALL WRITAPE (NT,IHD,NW,NOUT,NER)
GC TO 500

C 200 IRN = IR
IF (IR.LT.7) GO TO 250
IF (IR.GT.8) GO TO 210
IRN = IR+2*(IK-1)
GC TO 250

C 210 IRN = IR-2 + 2*DDIRL(15,J)

C 250 LRN = DDIRL(5,J)
NRSKP = IRN-LRN-1
IF (NRSKP.EQ.0) GO TO 300
IF (NRSKP.GT.0) GO TO 280
REWIND NT
WRITE (NOUT,2) NT
2 FORMAT (IHO,5(4H****)/5HOTAPE,I2,8H REWOUND/IHO,5(4H****) )
IF (IR.EQ.1) GO TO 300
NRSKP = IRN-1

C 280 DC 290 I=1,NRSKP
290 READ(NT)

C 300 CALL RDTAPE (NT,IHD,NW,NOUT,NER)
IF (NER.NE.0) GO TO 1000
DDIRL(4,J) = 1

C 500 IF (IR.GT.1) GO TO 550
IF (IHD(1).EQ.ITYP) GO TO 510
NEP = 1
WRITE (NOUT,501) ITYP, IHD(1)
501 FORMAT (IHO, INVALID FILE/IHO, FILE SHOULD BE = ,A4)
1 /14, FILE IS = ,A4)
GC TO 990

C 510 DC 520 I=3,12
520 DDIRL(I+7,J) = IHD(I)
IF (.NOT.PEAD) WRITE (NOUT,3) NT
IF (READ) WRITE (NOUT,4) NT
WRITE (NOUT,5) (IHD(I),I=2,12)
3 FORMAT (28H HEADER DATA WRITTEN ON TAPE,I2)
4 FORMAT (28H HEADER DATA READ FROM TAPE,I2)
5 FORMAT (5X5HIDENT,4X6HINTLO,4X6HNENGS,6X4HNSYM
1, 5X5HNASYM,3X2HNK
2, 4X6HNABGRP,4X6HNSBGRP,4X6HNPCXES,4X6HNAERSB
3 / 11110)

C 550 DDIRL(5,J) = IRN
GC TO 1000

```

WRUNI105
WRUNI106
WRUNI107
WRUNI108

C 990 DDTHL(20,J) = 1
1000 RETURN
END

```

SUBROUTINE CSDLM (INPUT)
C
C
C      DUMMY MAIN FOR THE DOUBLET-LATTICE PART OF THE VIBRA-4 PROGRAM
C
COMMON /AKOCOM/ NTL, MODES
X, NP, MSTRIP, NSMAX, NSMAX, NTOIAL, NB, MSBE, MBE
Y, ND, NE, NBY, NBZ, NTU, NTP, NTY, NIZ
1, NTYS, NTZS, MAXGR, MAXSTR, NSBETO, NSTRIP, KR, XM, RLFA, REFC
2, REFS, FMACH, LINES
C
COMMON NAA,A(1)
C
10 FORMAT (6I12)
C
NTL = INPUT
DATA SECT/4HSECT/, AERO/4HAERO/
C
C      FIND THE SECT OK AERO INPUT DATA
C
ISECT = 0
READ (NTI,90) CHECK
90 FORMAT (A4)
IF (CHECK.NE.SECT) GO TO 110
SECT INPUT DATA FOUND
ISECT = 1
READ (NTI,10) NMS, NUOF, NSYM, NASYM
C
100 READ (NTI,90) CHECK
110 IF (CHECK.NE.AERO) GO TO 100
C
C      AERO INPUT DATA FOUND
C
READ (NTI,10) NUDES, NSYM, NASYM, MFIX1, MFIX2
MDOFP = NUDES
MDOFB = NUDES
IF (ISECT.NE.0) MDOFP = 4*NUDES
IF (ISECT.NE.0) MDOFB = 2*NUDES
MODES = NSYM + NASYM
READ (NTI,10) NP, MSTRIP, NSMAX, NCMAX, NBOXES
NSMAX = NSMAX+1
NCMAX = NCMAX+1
READ (NTI,10) NB, MSBE, MBE
NTOTAL = NBOXES + 2*MAXO(MSBE, MBE)
MSBE = MSBE+1
MBE = MBE+1
C
L1 = NAA
L2 = L1 + NUDES
L3 = L2 + NUDES
L4 = L3 + MDOFP * MODES
L5 = L4 + MDOFB * MODES
L6

```

```

L7  + MDOF 3 * MODES
L8  NP * Nu
L9  NP * Nu
L10 NP
L11 NP
L12 NP
L13 MSTRIP
L14 MSTRIP
L15 2*NB
L16 2*NB
L17 NB
L18 NB
L19 NB
L20 NB
L21 NB
L22 NB
L23 NB
L24 NB
L25 MDE
L26 + Nb*10
L27 + MSBE
L28 MSBE
L29 + MSBE
L30 MSBE
L31 MSTRIP + NB
L32 MSTRIP + NB
L33 MSTRIP + NB
L34 MSTRIP + NB
L35 MSTRIP + NB
L36 MSTRIP + NB
L37 MSTRIP + NB
L38 MSTRIP + NB
L39 MSTRIP + NB
L40 MSTRIP + NB
L41 NCTAL
L42 NCTAL
L43 NCTAL
L44 NCTAL
L45 NCTAL
L46 NCTAL
L47 NCTAL
L47A = L47 + 4*NCTAL
L48 = L47A + 2*NCTAL
L49 = L48 + 2*NCTAL
L50 = L49 + 2*NCTAL
L51 = L50 + 2*(MAXO(NSYM,NASYM)+20)
NAA = L51
IF (ISECT.EQ.0) GO TO 200
ISECT = L51
NAA = L48 + NDOF*NMS*MODES

IPR = 1
CALL INPUTA (NTI,NYS,NDOF,MODES,IPR

```

CSOLM102
CSOLM103
CSOLM104
CSOLM105
CSOLM106
CSOLM107
CSOLM108
CSOLM109
CSOLM110
CSOLM111
CSOLM112
CSOLM113
CSOLM114
CSOLM115
CSOLM116
CSOLM117
CSOLM118
CSOLM119
CSOLM120
CSOLM121
CSOLM122
CSOLM123
CSOLM124
CSOLM125
CSOLM126
CSOLM127

```

1 , A(L1), A(L2), A(L3), A(L40) )
200 CONTINUE
      CALL SDLM (ELX1, LYL1, ELZ1, PHIN, PHIZ, PHIY, NODES
1 , NSYM, NASYM, MFIX1, MFLX2, ISECT, NDOF, PHI, MDOFP, MDOFB
2 , NAS, NASB, NCARAY, NSARAY, ISSIR, VSSIR
3 , IFLA, NBEA, NNT12, NSBEA
4 , VB, ZB, ARB, AVR, XLE, XIE, RIA, THIA, AO, AOP, XIS1, XIS2
5 , CU, CS, EG, SG, YS, ZS, XIJ, YIN, ZIN, CUORD
6 , X, XIC, DELX, XLAM, H, UHDX, DELA
7 , JT, DIA, RNS )
      CALL SDLM (A(L1), A(L2), A(L3), A(L4), A(L5), A(L6), NDOES
1 , NSYM, NASYM, MFIX1, MFLX2, ISECT, NDOF, A(L43), MDOFP, MDOFB
2 , A(L7), A(L8), A(L9), A(L10), A(L11), A(L12), A(L13)
3 , A(L14), A(L15), A(L16), A(L17)
4 , A(L18), A(L19), A(L20), A(L21), A(L22), A(L23)
5 , A(L24), A(L25), A(L26), A(L27), A(L28), A(L29), A(L30)
6 , A(L31), A(L32), A(L33), A(L34), A(L35), A(L36), A(L37), A(L38)
7 , A(L39), A(L40)
      A(L41), A(L42), A(L43), A(L44), A(L45), A(L46), A(L47)
      A(L48), A(L49), A(L50), A(L51), A(L52), A(L53), A(L54), A(L55),
      A(L56), A(L57), A(L58), A(L59), A(L60), A(L61), A(L62), A(L63),
      A(L64), A(L65), A(L66), A(L67), A(L68), A(L69), A(L70), A(L71),
      A(L72), A(L73), A(L74), A(L75), A(L76), A(L77), A(L78), A(L79),
      A(L80), A(L81), A(L82), A(L83), A(L84), A(L85), A(L86), A(L87),
      A(L88), A(L89), A(L90), A(L91), A(L92), A(L93), A(L94), A(L95),
      A(L96), A(L97), A(L98), A(L99), A(L100), A(L101), A(L102), A(L103),
      A(L104), A(L105), A(L106), A(L107), A(L108), A(L109), A(L110),
      A(L111), A(L112), A(L113), A(L114), A(L115), A(L116), A(L117),
      A(L118), A(L119), A(L120), A(L121), A(L122), A(L123), A(L124),
      A(L125), A(L126), A(L127)
      RETURN
      END

```

```

SUBROUTINE AERO (MFI1, MFI2, NG, NMSYM, NTO, NOUT, NM, NEWBFZ,
1  NEWREV, NSTOT, CRSPAN, DCP, FZ, FY, CN, CM, SPLD,
2  ISSTR, NSBEA, NBARAY, NCARAY, YB, ZB, XIS1, XIS2, CG, CS, EE, SG, YS, ZS,
3  XIC, XIJ, DELX, COORD, CZB, CYB, CNB, CMB, CPR, CPI,
4  DIMENSION ISSTR(1), NSBEA(1), NBARAY(1), NCARAY(1), YB(1), ZB(1),
5  DIMENSION XIS1(1), XIS2(1), CG(1), CS(1), EE(1), SG(1), YS(1), ZS(1),
6  DIMENSION XIC(1), XIJ(1), DELX(1), COORD(1)
7  ** COMPUTES AERODYNAMIC PARAMETERS FOR ALL LIFTING SURFACES
8  AND ALL SLENDER BODIES
9  COMMON /AROCOM/ NTI, MODES
10 X, NP, NSTRI, NSMAX, NCMAX, NTO, NTP, NTY, NTZ
11 Y, ND, NE, NBY, NBZ, NTO, NTP, NTY, NTZ
12 I, NTYS, NTZS, MAXGR, MAXSTR, NSBELO, NSTRI, KR, XM, REFA, REFC
13 REFS, FMACH, LINES
14 COMPLEX DCP(1), FZ(1), FY(1), CN(1), CM(1), SPLD(1)
15 COMPLEX CZB(1), CYB(1), CNB(1), CMB(1)
16 COMPLEX CZT, CYT, CMT, CNT, CXT, CLT, FZLB, FYLB
17 COMPLEX CZTS, CYTS, CMTS, CNTS, CLTS
18 REAL KR
19 10 FORMAT (1H1, 10X, 21H*** PRESSURES *** //)
20 13X, 4HCR = F8.4 //)
21 10H PRESSURES //)
22 10H PRESSURES //)
23 10H PRESSURES //)
24 10H PRESSURES //)
25 10H PRESSURES //)
26 10H PRESSURES //)
27 10H PRESSURES //)
28 10H PRESSURES //)
29 10H PRESSURES //)
30 10H PRESSURES //)
31 10H PRESSURES //)
32 10H PRESSURES //)
33 10H PRESSURES //)
34 10H PRESSURES //)
35 10H PRESSURES //)
36 10H PRESSURES //)
37 10H PRESSURES //)
38 10H PRESSURES //)
39 10H PRESSURES //)
40 10H PRESSURES //)
41 10H PRESSURES //)
42 10H PRESSURES //)
43 10H PRESSURES //)
44 10H PRESSURES //)
45 10H PRESSURES //)
46 10H PRESSURES //)
47 10H PRESSURES //)
48 10H PRESSURES //)
49 10H PRESSURES //)

```

```

100 FORMAT ( //35X, 4HMODE, I3//25X, 26HTOTALS ON LIFTING SURFACES )
110 FCFORMAT ( //10X, 7HC2 =, 2F15.6, 10X, 7HCY =, 2F15.6 / 10X,
1 7HCM =, 2F15.6, 10X, 7HCN =, 2F15.6 / 10X,
2 7HCSL =, 2F15.6 / )
120 FCFORMAT ( //25X, 27HTOTALS ON ENTIRE AIRCRAFT )
130 FCFORMAT ( //15, F12.4, 3X, 4F14.5 )
140 FCFORMAT ( //1H1 )
150 FCFORMAT ( //48H STATION SPANWISE SPAN LOAD (C*CL)/(C-BAR)
1 WRITE (NOUT,10) KR REAL IMAG. /)

C
REWIND NW K = 1, MFIX2
DC 180 K = 1, MFIX2
READ (NW) (DCP(IK), IK=1,NIC1)
IF (K.NE.MFIX1 .AND. K.NE.MFIX2 ) GO TO 180
KK = K
IF (K.GT.NMSYM) KK = K-NG
WRITE (NOUT,20) KK
I2 = 0
LP = 1
LPAGE = 1
DO 170 J = 1, NSTRIIP
I1 = 12 + 1
I2 = 12 + NCARAY(LP)
DO 160 I = 1, I2
XCC = (XIC(I)-XIJ(J)) / CS(J)
WRITE (NOUT,70) LP, J, I, XCC, XIC(I), YS(J), ZS(J), DCP(I)
IF (LPAGE*LNES.NE.I) GC TO 160
LPAGE = LPAGE + 1
WRITE (NOUT,20) KK
160 CONTINUE
IF (I2.EQ.NBARAY(LP)) LP=LP+1
170 CONTINUE
180 CONTINUE

C ***
C
C
SYMA = 2.0
SYMR = 0.0
DC 190 I=1,NSTCT
CPI(I) = 0.0
CPI(I) = 0.0
FZ(I) = (0.0,0.0)
FY(I) = (0.0,0.0)
190 CONTINUE
WRITE (NOUT,30)
IF (NR.EQ.0) GO TO 210
WRITE (NOUT,40)
REWIND NEWRFZ
REWIND NEWRFY
210 CONTINUE
REWIND NW

```

AERO 50
AERO 51
AERO 52
AERO 53
AERO 54
AERO 55
AERO 56
AERO 57
AERO 58
AERO 59
AERO 60
AERO 61
AERO 62
AERO 63
AERO 64
AERO 65
AERO 66
AERO 67
AERO 68
AERO 69
AERO 70
AERO 71
AERO 72
AERO 73
AERO 74
AERO 75
AERO 76
AERO 77
AERO 78
AERO 79
AERO 80
AERO 81
AERO 82
AERO 83
AERO 84
AERO 85
AERO 86
AERO 87
AERO 88
AERO 89
AERO 90
AERO 91
AERO 92
AERO 93
AERO 94
AERO 95
AERO 96
AERO 97
AERO 98
AERO 99
AERO 100
AERO 101

AERO 102
 AERO 103
 AERO 104
 AERO 105
 AERO 106
 AERO 107
 AERO 108
 AERO 109
 AERO 110
 AERO 111
 AERO 112
 AERO 113
 AERO 114
 AERO 115
 AERO 116
 AERO 117
 AERO 118
 AERO 119
 AERO 120
 AERO 121
 AERO 122
 AERO 123
 AERO 124
 AERO 125
 AERO 126
 AERO 127
 AERO 128
 AERO 129
 AERO 130
 AERO 131
 AERO 132
 AERO 133
 AERO 134
 AERO 135
 AERO 136
 AERO 137
 AERO 138
 AERO 139
 AERO 140
 AERO 141
 AERO 142
 AERO 143
 AERO 144
 AERO 145
 AERO 146
 AERO 147
 AERO 148
 AERO 149
 AERO 150
 AERO 151
 AERO 152
 AERO 153

```

212 CCNTINUE
    DC 260 J=1,NSTRIP
    I1 = I2+1,NCARAY(LP)
    I2 = I2+1,NCARAY(LP) LP=LP+1
    XI145 = 0.25*CS(J) + XIJ(J)
    CN(J) = (0.0,0.0)
    CM(J) = (0.0,0.0)
    SPLN(J) = 0.0
    DC 220 I=1,I2
    CN(J) = CN(J) + DCP(I)*DELX(I)
    CM(J) = CM(J) + DCP(I)*DELX(I)*(XIC(I) - XI145)
220 CCNTINUE
    YCS = YS(J)/REFS
    CH2 = CS(J)*2
    JL(JL) = SPLD(JL) + CN(J) / CBSPAN
    CN(J) = CN(J)/CS(J)
    CM(J) = -CM(J)/CH2
    ARSCNJ = SQRT((REAL(CN(J)))**2 + (AIMAG(CN(J)))**2)
    IF (ABS(REAL(CN(J))) - LE, 0.000001) GO TO 222
    IF (ARSCNJ - LE, 0.00001) GO TO 230
    CPR(J) = -REAL(CM(J))/REAL(CN(J)) + 0.25
    GO TO 224
    CPO(J) = 0.0
222 CCNTINUE
224 IF (ABS(AIMAG(CN(J))) - LE, 0.0000001) GO TO 240
    CPI(J) = -AIMAG(CM(J))/AIMAG(CN(J))*0.25
    GO TO 250
230 CCNTINUE
    CPR(J) = 0.0
240 CCNTINUE
    CPI(J) = 0.0
250 CCNTINUE
260 WRITE (NOUT, 72) J,YS(J), ZS(J), YOS,CN(J),CM(J),CPR(J),CPI(J)
    CCNTINUE
    WRITE (NOUT,140)
    WRITE (NOUT,150)
    DC 270 JL=1, MAXSTR
    WRITE (NOUT,130) JL, COORD(JL), SPLD(JL)
  
```

```

270 CONTINUE
271 CONTINUE
IF (NTZS.EQ.0. AND .NTYS.EQ.0) GO TO 300

READ (NEWREFZ) (FZ(LL),
READ (NEWBFY) (FY(LL),
IF (K .NE. MFIX1 .AND. K .NE. MFIX2) GO TO 390
L2 = 0
DC 290 N=1, NR
CZR(N) = (0.0,0.0)
CYR(N) = (0.0,0.0)
CMR(N) = (0.0,0.0)
CNR(N) = (0.0,0.0)
L1 = L2+1
L2 = L2 + NSBEA(N)
SRL = XIS2(L2) - XIS1(L1)
WRITE (NCUT,60) KK
DC 280 LB=L1,L2
LX = LX+1
XSR = 0.5*(XIS1(LB) + XIS2(LB))
DXSR = XIS2(LB) - XIS1(LB)
XCI = (XSB - XIS1(L1))/ SRL
F7LB = FZ(LB)/ DXSB
F7LR = FY(LB)/ DXSB
WRITE (NCUT,72) LB, YB(N), ZR(N), XOL, F7LB, F7LR
CZB(N) = CZR(N) + FZ(LB)
CYB(N) = CYR(N) + FY(LB)
CMB(N) = CMB(N) - F7(LB)*(XSB-XIS1(L1))
CNB(N) = CNB(N) - F7(LB)*(XSR-XIS1(L1))

CONTINUE
CZB(N)/REFA
CYB(N)/REFA
CMR(N)/(REFA*REFC)
CNR(N)/(REFA*REFC)
XCEN = (XIS1(L1) + XIS2(L2))/2.0
WRITE (NCUT,90) N, CZB(N), CMB(N), CYB(N), CNB(N)

280 CONTINUE
CZB(N)/REFA
CYB(N)/REFA
CMR(N)/(REFA*REFC)
CNR(N)/(REFA*REFC)
XCEN = (XIS1(L1) + XIS2(L2))/2.0
WRITE (NCUT,90) N, CZB(N), CMB(N), CYB(N), CNB(N)

290 CONTINUE
300 CONTINUE
WRITE (NCUT,140)
C7T = (0.0,0.0)
C7Y = (0.0,0.0)
CMT = (0.0,0.0)
CAT = (0.0,0.0)
CCT = (0.0,0.0)
DC 320 J=1, NSTRIP
CH2 = CS(J)*2
XI14S = 0.25*CS(J) + XIJ(J)
CMULT = 2.0*FE(J)*CG(J)
SMULT = 2.0*FE(J)*SG(J)
GUCJ = 1.0
IF (ABS(YS(J)).LE.0.0001 .AND. ABS(CG(J))*CN(J)*(XI14S-XM)
CXT = CH2*CM(J) - CS(J)*CN(J) GUCJ=0.5

```

AERO 154
AERO 155
AERO 156
AERO 157
AERO 158
AERO 159
AERO 160
AERO 161
AERO 162
AERO 163
AERO 164
AERO 165
AERO 166
AERO 167
AERO 168
AERO 169
AERO 170
AERO 171
AERO 172
AERO 173
AERO 174
AERO 175
AERO 176
AERO 177
AERO 178
AERO 179
AERO 180
AERO 181
AERO 182
AERO 183
AERO 184
AERO 185
AERO 186
AERO 187
AERO 188
AERO 189
AERO 190
AERO 191
AERO 192
AERO 193
AERO 194
AERO 195
AERO 196
AERO 197
AERO 198
AERO 199
AERO 200
AERO 201
AERO 202
AERO 203
AERO 204
AERO 205

AERO 258
 AERO 259
 AERO 260
 AERO 261
 AERO 262
 AERO 263
 AERO 264
 AERO 265
 AERO 266
 AERO 267
 AERO 268

```

380 CCNTI NUC
    CZT = SYMA * CZT
    CYT = SYMB * CYT
    CMT = SYMA * CMT
    CNT = SYMB * CNT
    CLT = SYMR * CLT
    WRITE (NOUT,120)
    WRITE (NOUT,110)
390 CCNTI NUC
    PFTURN
    END

    / (2.0*REFS*REFA)
    CZT , CYT , CMT , CNT , CLT
  
```

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 ATAN3

```

SUBROUTINE ATAN3 (Y,X,T)
**#   CCMPTUES   T=ATAN(Y/X)
PI = 3.141593
PI2 = PI*2.0
IF (X.EQ.0.) T=PI/2.
IF (X.EQ.0.) GO TO 2
IF (Y.EQ.0.) T=0.
IF (Y.EQ.0.) GO TO 2
A = Y/X
T = ATAN(A)
2 CONTINUE
IF ((Y.GE.0.).AND.(X.EQ.0.)) GO TO 1
IF ((Y.EQ.0.).AND.(X.GT.0.)) T=0.
IF ((Y.EQ.0.).AND.(X.GT.0.)) GO TO 1
IF ((Y.GE.0.).AND.(X.LE.0.)) T=PI + T
IF ((Y.GE.0.).AND.(X.LE.0.)) GO TO 1
IF ((Y.LE.0.).AND.(X.LE.0.)) T=PI + T
IF ((Y.LE.0.).AND.(X.LE.0.)) GO TO 1
IF ((Y.LE.0.).AND.(X.GE.0.)) T=PI2+ T
IF ((Y.LE.0.).AND.(X.GE.0.)) GO TO 1
1 CONTINUE
RETURN
END
  
```

55


```

ICOL = ICOL + 1
GO TO 180
30 CONTINUE =
AREA =
BFS(ICOL, 1) = WIDTH * DELX (IBOX) * AREA
BFS(1COL,1) = BFS(1COL,1) * AREA
BFS(1COL,2) = BFS(1COL,2) * AREA
BFS(1COL,2) = BFS(1COL,2) * AREA
40 CONTINUE
50 CONTINUE
60 CONTINUE

- Z - ORIENTED BODIES AS SENDING ELEMENTS

70 CONTINUE
SGS = 0.0
CGS = 1.0
NASD = 0
IZYFLG = 1
ASSIGN 10 TO IBODY
ASSIGN 100 TO ICOLM
GO TO 80

*** LOOP FOR EACH INTERFERENCE BODY SENDING ELEMENT

80 CONTINUE = NTP
INDEX =
DO 130 ISB = 1, NB
  IF (NREA(2,ISB).EQ.2) GO TO 90
  IF (NREA(2,ISB).NE.IZYFLG) GO TO 120
90 DYS = NREA(1,ISB)
  NSRE = 1
  JPL = 1
  LAST = .FALSE.
  DZS = ZR (ISB)
  -- ISB -- IS THE ELEMENT OF THE SEND. BODY
  FARG2 = 1.0
  DO 110 ISRE = 1, NSRE
    FARG1 = EARG2
    INDEX = INDEX + 1
    DXS = X (INDEX) - DELX (INDEX) / 4.0
    FARG2 = KR * DELX (INDEX) / CBRAR
    CALCULATE THIS COLUMN
    ICOL = ICOL + 1
    BFS(1COL,1) = 0.0
    BFS(1COL,1) = 0.0
    BFS(1COL,2) = 0.0
    BFS(1COL,2) = 0.0
    FIKJ1 = CMPLX ( COS (EARG1), -SIN (EARG1) )
    FIKJ2 = CMPLX ( COS (EARG2), SIN (EARG2) )
  
```

```

C      GO TO 140
100 CONTINUE
C      IF ( ISRE .EQ. 1 ) GO TO 110
      BFS(ICOL-1,1) = BFS(ICOL-1,1)*EIKJ1 - BFS(ICOL,1)*EIKJ2
      BFS(ICOL-1,1) = BFS(ICOL-1,1)*EIKJ1 - BFS(ICOL,1)*EIKJ2
      BFS(ICOL-1,2) = BFS(ICOL-1,2)*EIKJ1 - BFS(ICOL,2)*EIKJ2
      BFS(ICOL-1,2) = BFS(ICOL-1,2)*EIKJ1 - BFS(ICOL,2)*EIKJ2
110 CONTINUE
      GO TO 130
120 CONTINUE
      INDEX = INDEX + NBEA(1,ISB)
130 CONTINUE
      RETURN TO CALLING POINT - EITHER Y OR Z SENDING BODY ELEMENT
C      *** GO EITHER TO THE Y-ORIENTED INTERFERENCE BODY ELEMENT LOOP
C      OR TO THE LOOP FOR SLENDER BODY SENDING ELEMENTS
C      GO TO 1BODY, (10,190)
C      CALCULATE EACH ROW OF THE SENDING COLUMN
C-----
140 CONTINUE
      IY = 0
      IPW = 0
      DC 170 IRB = 1, NB
      NRBE = NSBEA(IRB)
      XYB = YR( IRB )
      XZR = ZR( IRB )
      -- IRB -- IS THE RECEIVING BODY
      DC 160 IRBE = 1, NRBE
      IY = IY + 1
      IPW = IPW + 1
      DRIA = AO( IY )
      DXLE = XIS1( IY )
      DXTE = XIS2( IY )
      XX1 = DXLE
      XX2 = DXTE
      XAA = DRIA
      ICOL TO 20
      GO TO 20
150 CONTINUE
      CALL RWREC( IFLAG, IO , BFS( 1,1), LENGTH , 1, 0 )
      CALL RWREC( IFLAG, IO , BFS( 1,2), LENGTH , 1, 0 )
      CALL RWREC( IFLAG, IOA , BFS( 1,1), LENGTH , 1, 0 )
      CALL RWREC( IFLAG, IOA , BFS( 1,2), LENGTH , 1, 0 )
160 CONTINUE

```

BFSMA158
 BFSMA159
 BFSMA160
 BFSMA161
 BFSMA162
 BFSMA163
 BFSMA164
 BFSMA165
 BFSMA166
 BFSMA167
 BFSMA168
 BFSMA169
 BFSMA170
 BFSMA171
 BFSMA172
 BFSMA173
 BFSMA174
 BFSMA175
 BFSMA176
 BFSMA177
 BFSMA178
 BFSMA179
 BFSMA180
 BFSMA181
 BFSMA182
 BFSMA183
 BFSMA184
 BFSMA185
 BFSMA186
 BFSMA187
 BFSMA188
 BFSMA189
 BFSMA190
 BFSMA191
 BFSMA192
 BFSMA193
 BFSMA194
 BFSMA195
 BFSMA196
 BFSMA197
 BFSMA198
 BFSMA199
 BFSMA200
 BFSMA201
 BFSMA202
 BFSMA203
 BFSMA204
 BFSMA205
 BFSMA206
 BFSMA207
 BFSMA208
 BFSMA209

BF SMA262
BF SMA263
BF SMA264
BF SMA265
BF SMA266
BF SMA267
BF SMA268
BF SMA269

SCS = -1.0
CGS = 0.0
GC TO 200
250 CONTINUE
C
C GC TO 150
END

```

1  SUBROUTINE CTLS(NFAASS, NDAS, ND, FLXI, ELYI, ELZI, XA, YA, ZA,
2  XA, YB, ZB, XAR, YAR, ZAR, XBR, YBR, ZBR, NTP 6
3  XI, ETA,
4  )
5  THIS SUBROUTINE COMPUTES THE XI- AND ETA-COORDINATES OF A
6  *GENERATED NODE* ON ELASTIC AXIS OF A CONTROL SURFACE (FOR
7  WHICH NO INPUT NODE EXISTS) FROM THE ASSOCIATED NODE (NODASS)
8  OF THE ASSOCIATED ELASTIC AXIS (NEAASS)
9
10 DIMENSION ELXI(1), ELYI(1), FLZI(1)
11 DIMENSION XAR(1), YAR(1), ZAR(1), XBR(1), YBR(1), ZBR(1)
12 DIMENSION XI(1), ETA(1)
13
14 = XAR(NEAASS)
15 = YAR(NEAASS)
16 = ZAR(NEAASS)
17 = XBR(NEAASS)
18 = YBR(NEAASS)
19 = ZBR(NEAASS)
20
21 AXI = FLXI(NDAS)
22 AETA = ELYI(NDAS)
23 AZET = ELZI(NDAS)
24
25 YDIF = YR - YA
26 ZDIF = ZR - ZA
27 RHO = SQRT(YDIF**2 + ZDIF**2)
28 YDIF = YDIF / RHO
29 ZDIF = ZDIF / RHO
30
31 YDIF = YDIF * COSG + ZDIF * SING
32 (AYA - YA) * COSG + (AZA - ZA) * SING
33 (AYB - YB) * COSG + (AZB - ZB) * SING
34 (AETA - YA) * COSG + (AZET - ZA) * SING
35
36 AXB = AXI / (AYBW - AETAW)
37 AXB = AXB - AXA / (AYBW - AETAW)
38
39 TEST1 = ABS(AXB - AETAW)
40 IF (TEST1 .GE. 0.01) GO TO 10
41
42 TEST2 = ABS(AXI - AMA * AETAW - AXA + AMAP * AYAW)
43 IF (TEST2 .LT. 0.01) GO TO 30
44
45 CONTINUE
46 WRITE (NTP, 20)
47
48 FORMAT (/// 10X, 70H*** WARNING *** NODAL POINT(S) DO NOT
49 LIE EXACTLY ON ELASTIC AXIS
50 // )
51
52 CONTINUE
53 AYB = AYA
54 AZB = AZA
55 AHO = SQRT(AYDIF**2 + AZDIF**2)
56 APOSS = AYDIF / ARHO

```

```

      ASING      = AZDIF / ARHO
      TEST3     = ARS(ASING - SING)
      TEST4     = ARS(ACOSG - COSG)
      IF (TEST3 .GT. 0.001 .OR. TEST4 .GT. 0.001) WRITE (NTP6,70)

      AETYDF    = AFTAW - YAB
      AM        = (XB - XA) / (YBR - YAB)
      DENOM     = 1.0 + AM*AMA

      XI(ND)    = (AM*AETYDF + XA + AM*AMA*AXI) / DENOM
      ETA(ND)   = (AMA*(XI - XA) + AETAW + AM*AMA*YAB) / DENOM

      WRITE (NTP6,40) ND, NEAASS, NDAS, XA, YAB, XB, YBR
      WRITE (NTP6,50) AXA, AVA, AZA, AXB, AYB, AZB, AXI, AETA, AZET
      WRITE (NTP6,60) AMA, AMAP, AM, AETAW, AETYDF, XI(ND), ETA(ND)

      40 FORMAT ( // 4X, 38HND, NEAASS, NDAS, XA, YAB, XB, YBR /
      1 3I6, 4E16.6 / )
      50 FORMAT ( // 6X, 28HAXI, AYA, AZA, AXB, AYB, AZB / 6F12.5 /
      1 3I6, 28HAXI, AETA, AZET / 6F12.5 / )
      60 FORMAT ( // 6X, 28HAMA, AMAP, AM, AETAW, AETYDF / 5F12.5 /
      1 3I6, 28HAXI(ND), ETA(ND) / 5F12.5 / )
      70 FORMAT ( /// 10X, 79H*** WARNING *** THE TWO ELASTIC AXES DO NOT LIE EXACTLY IN THE SAME PLANE /// )
      1 10X, 79H*** WARNING *** THE TWO ELASTIC AXES DO NOT LIE EXACTLY IN THE SAME PLANE /// )
      RETURN
      END

```

C	SUBROUTINE D01P (N1PH,NTPDH,N1PH4,MASTH,MASTDH,MASTH4,IFNEWH, 1 NSYM,NASYM,NODES,MODES,NB,IRW, 2 PH17,PH17,PH17,COL,WORK)	D01P	2
C	DIMENSION PHIN(NODES, MODES), PH17(NODES, MODES), 1 PH17(NODES,MODES), COL(IRW), WORK(1)	D01P	3
C	REWIND NTPH REWIND NTPDH	D01P	4
C	IF (NB.EQ.0) GO TO 70 MASTAP = MASTH NTAPE = NTPH IGD = 1 10 REWIND NTAPE	D01P	5
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	6
C	GO TO 110	D01P	7
C	30 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 2 GO TO 10	D01P	8
C	40 CONTINUE MASTAP = MASTH NTAPE = NTPH IGD = 3 50 REWIND NTAPE	D01P	9
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	10
C	GO TO 110	D01P	11
C	60 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 4 GO TO 50	D01P	12
C	70 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 5 80 CONTINUE	D01P	13
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	14
C	GO TO 110	D01P	15
C	90 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 6 GO TO 60	D01P	16
C	100 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 7 110 CONTINUE	D01P	17
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	18
C	GO TO 110	D01P	19
C	120 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 8 GO TO 90	D01P	20
C	130 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 9 140 CONTINUE	D01P	21
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	22
C	GO TO 110	D01P	23
C	150 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 10 GO TO 120	D01P	24
C	160 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 11 170 CONTINUE	D01P	25
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	26
C	GO TO 110	D01P	27
C	180 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 12 GO TO 150	D01P	28
C	190 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 13 200 CONTINUE	D01P	29
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	30
C	GO TO 110	D01P	31
C	210 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 14 GO TO 180	D01P	32
C	220 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 15 230 CONTINUE	D01P	33
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	34
C	GO TO 110	D01P	35
C	240 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 16 GO TO 210	D01P	36
C	250 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 17 260 CONTINUE	D01P	37
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	38
C	GO TO 110	D01P	39
C	270 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 18 GO TO 240	D01P	40
C	280 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 19 290 CONTINUE	D01P	41
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	42
C	GO TO 110	D01P	43
C	300 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 20 GO TO 270	D01P	44
C	310 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 21 320 CONTINUE	D01P	45
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	46
C	GO TO 110	D01P	47
C	330 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 22 GO TO 300	D01P	48
C	340 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 23 350 CONTINUE	D01P	49
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	50
C	GO TO 110	D01P	51
C	360 CONTINUE MASTAP = MASTDH NTAPE = NTPDH IGD = 24 GO TO 330	D01P	52
C	370 CONTINUE IF (IFNEWH.NE.0) RETURN MASTAP = MASTH NTAPE = NTPH IGD = 25 380 CONTINUE	D01P	53
C	CALL ORGN (NTAPE,MASTAP,NSYM,NASYM,NODES,MODES,IRW, 1 COL,PH17,WORK)	D01P	

54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77

D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P
 D01P

```

C      1 COL,PHIN,WORK )
      90 GO TO 110
        CONTINUE = MASTDH
        NTAPF = NTPDH
        IGO = 6
      100 GO TO 80
        CONTINUE = MASTH4
        NTAPF = NTPH4
        REWIND NTAPF
        IGO = 7
      110 CONTINUE
C
      120 GO TO (30, 40, 60, 70, 90, 100, 120), IGO
        CONTINUE MASTH
        REWIND MASTDH
        REWIND MASTH4
        RETURN
      END
  
```

```

SURROUTINE DPPS(K,KS,I,J1,J2,SGR,CGR,REFC,FMACH,YS,ZS,NBARAY,
1 NCARAY,DT,DTA
2 YB,ZB,ARB,AVR,XLF,XTE,X,CG,EE,SG,XIC,DELX,XLAM, SCALER )
DIMENSION YB(1), ZB(1), ARB(1), AVR(1), XLE(1), XTE(1), X(1)
DIMENSION CG(1), EE(1), SG(1), XIC(1), DELX(1)
DIMENSION XLAM(1)
*** GENERATES ROWS OF THE DPP SUBMATRIX USING
SUBROUTINE SUBP
REAL KR
DIMENSION YS(1), ZS(1), NBARAY(1), NCARAY(1)
COMPLEX SUM(2), DT(1), DTA(1)
BETA = SORT(1.0-FMACH**2)
FL = REFC
L = 1
C L IS THE PANEL NUMBER ASSOCIATED WITH SENDING POINT J
C LS IS THE STRIP NUMBER ASSOCIATED WITH SENDING POINT J
IC = 0
NPXS = NBARAY(L)
NCI = NCARAY(L)
NCPNR = NCI
NRCUM = NCI
YREC = YS(KS)
ZREC = ZS(KS)
DO 20 J=J1,J2
IC = IC+1
IR = J
CALL SURP (I,L,LS,J,IC,IR,NBXS,NCPNR,SGR,CGR,YREC,ZREC,SUM
1 , NCARAY,YB,ZB,ARB,AVR,XLE,XTE,X,CG,EE,SG,XIC,DELX,XLAM )
2 , SCALER
DT(J)=SUM(1)
DTA(J)=SUM(2)
IF (J.EQ.J2) GO TO 20
IF (IC.EQ.NCI) IC=0
IF (J.LT.NBXS) GO TO 10
L = L+1
NCI = NCARAY(L)
NCPNR = NBXS + NCI
NPXS = NBARAY(L)
10 CONTINUE
IF (J.LT.NBXCUM) GO TO 20
LS = LS+1
NRCUM = NRCUM+NCI
20 CONTINUE
RETURN
END

```



```

30 CONTINUE
  THETA = TH1A(IX)
  THPI = TH1A(IXPI)
  THMI = TH1A(IXMI)
40 CONTINUE
  IF (IX.EQ.1X1) THM1=THM1-2.0*PI
  IF (IX.EQ.1X2) THPI=THPI+2.0*PI
  DELTH = 0.5*(THPI - THMI)
  YREC = YR(KB)+AVR(KB)*COS(THETA)
  ZREC = ZR(KB)+AVR(KB)*SIN(THETA)
  RHO = SQRT(1.0+(ARR(KB)**2 - 1.0) * (COS(THETA))**2)
  SCR = -ARR(KB)*COS(THETA)/RHO
  CCR = SIN(THETA)/RHO
  CMULT = SIN(THETA) * RHO / PI
  CMULT = COS(THETA) * RHO / PI
  DO 90 J=J1,J2
  IF J = J1+1
  IF J = J
  CALL SURP (I,L,LS,J,IO,IR,NPXS,NCPNB,SGR,CGR,YREC,ZREC,SUM
1 , NCARAY,YB,ZB,APB,AVR,XLE,XTE,X,CG,FE,SG,Y S,ZS,XIC,DELX,XLAM )
2 , SCALER
  GO TO (50,50,60), NZYKB
50 CONTINUE
  OP7(J) = OPZ(J) + SUM(1) * SMULT * DELTH
  OP7A(J) = OP7A(J) + SUM(2) * SMULT * DELTH
  IF (NZYKB.EQ.1) GO TO 70
60 CONTINUE
  OPY(J) = OPY(J) + SUM(1) * CMULT * DELTH
  OPYA(J) = OPYA(J) + SUM(2) * CMULT * DELTH
70 CONTINUE
  IF (J.EQ.J2) GO TO 90
  IF (IO.EQ.NCI) IO=0
  IF (J.LT.NBXS) GO TO 80
  L = L+1
  NCI = NCARAY(L)
  NCPNB = NPXS + NCI
  NBXS = NBARAY(L)
80 CONTINUE
  IF (J.LT.NRCUM) GO TO 90
  LS = LS+1
  NRCUM = NRCUM+NCI
90 CONTINUE
100 CONTINUE
  RETURN
  END

```

DPZY 54
 DPZY 55
 DPZY 56
 DPZY 57
 DPZY 58
 DPZY 59
 DPZY 60
 DPZY 61
 DPZY 62
 DPZY 63
 DPZY 64
 DPZY 65
 DPZY 66
 DPZY 67
 DPZY 68
 DPZY 69
 DPZY 70
 DPZY 71
 DPZY 72
 DPZY 73
 DPZY 74
 DPZY 75
 DPZY 76
 DPZY 77
 DPZY 78
 DPZY 79
 DPZY 80
 DPZY 81
 DPZY 82
 DPZY 83
 DPZY 84
 DPZY 85
 DPZY 86
 DPZY 87
 DPZY 88
 DPZY 89
 DPZY 90
 DPZY 91
 DPZY 92
 DPZY 93
 DPZY 94
 DPZY 95
 DPZY 96
 DPZY 97
 DPZY 98

```

SUBROUTINE DUMULT ( N1 , N2 , NTZS , NTYS ,
DUMULT 3
DUMULT 4
DUMULT 5
DUMULT 6
DUMULT 7
DUMULT 8
DUMULT 9
DUMULT 10
DUMULT 11
DUMULT 12
DUMULT 13
DUMULT 14
DUMULT 15
DUMULT 16
DUMULT 17
DUMULT 18
DUMULT 19
DUMULT 20
DUMULT 21
DUMULT 22
DUMULT 23
DUMULT 24
DUMULT 25
DUMULT 26
DUMULT 27
DUMULT 28
DUMULT 29
DUMULT 30
DUMULT 31
DUMULT 32
DUMULT 33
DUMULT 34
DUMULT 35
DUMULT 36
DUMULT 37
DUMULT 38
DUMULT 39
DUMULT 40
DUMULT 41
DUMULT 42
DUMULT 43
DUMULT 44
DUMULT 45
DUMULT 46
DUMULT 47

C DUMULT
C SUBROUTINE DUMULT ( N1 , N2 , NTZS , NTYS ,
C DZ MATRIX MULTIPLICATION
C W = DZ * UZ + DY * UY
C
C ***
C N1 FIRST ROW OF W TO CALCULATE
C N2 LAST ROW OF W TO CALCULATE
C NTZS NO. COL. IN DZ AND NO. ROWS
C NTYS NO. COL. IN DY AND NO. ROWS
C W OUTPUT COLUMN (ROWS N1 THRU N2 )
C DZ MATRIX (ROWS N1 THRU N2 )
C UZ MATRIX
C DY MATRIX
C UY MATRIX
C
C ***
C COMPLEX W(1) , DZ (NTZS, 1) , UZ ( NTZS ) ,
C DY ( NTYS, 1 ) , UY ( NTYS )
C
C IF ( NTZS .LE. 0 ) GO TO 100
C K = 0
C DO 20 I = N1, N2
C K = K + 1
C DO 10 J = 1, NTZS
C W(I) = W(I) + DZ( J, K ) * UZ( J )
C 10 CONTINUE
C 20 CONTINUE
C
C 100 IF ( NTYS .LE. 0 ) GO TO 200
C W = W + DY * UY
C K = 0
C DO 120 I = N1, N2
C K = K + 1
C DO 110 J = 1, NTYS
C W(I) = W(I) + DY( J, K ) * UY( J )
C 110 CONTINUE
C 120 CONTINUE
C 200 RETURN
C END

```

```

1 SUBROUTINE DYPZ(KB,KS,LS,IZ,I,J1,J2,NYFLAG,FLND,FLNF,SGR,CGR,REFC,DYPZ
2 FMACH, KR, ARB, NBEA, LBO, LS0, J80, DT, DTA
3 YB, ZB, RIA, X, YS, ZS, DELX )
4 DIMENSION YR(1), ZB(1), RIA(1), X(1), YS(1), ZS(1), DELX(1)
5 *** GENERATES ROWS OF THE SUBMATRICES DYP, OYZ AND DYY
6 USING SURROUTINE SUBB
7 KR, M
8 REAL SUM(2), DT(1), DTA(1)
9 COMPLEX SUM(2,1), ARB(1)
10 DIMENSION NBEA(2,1), ARB(1)
10 FFORMAT (1H0,6E20.8)
11 NDY = 1
12 NYFL = 0
13 NYFLAG = 0
14 IMN = 0
15 PI = 3.1415926
16 FPS = 0.00001
17 BETA = SQRT(1.0-FMACH**2)
18 FL = REFC
19 M = FMACH
20 NRV = 0
21 JZ = 0
22 LR = LBO
23 IS THE BODY NUMBER ASSOCIATED WITH SENDING POINT J
24 SGR = -1.0
25 CGS = 0.0
26 LS = LSC
27 LS IS THE INDEX OF THE Y AND Z COORDINATES OF SENDING POINT J
28 LS RUNS FROM NSTRIP+NB-NBY+1 THROUGH NSTRIP+NB
29 JZ = JZ-1
30 NZYLR = NBEA(2,LB)
31 AR = APH(LB)
32 SI = 0.0
33 CL = 1.0
34 TL = 0.0
35 DR = 20
36 J = J1, J2
37 JR = 1
38 JZ = JZ+1
39 CALL SUBB(KB,KS,I,J,JZ,J8,L8,LS,NDY,NYFL,FLND,FLNF,PI,EPS,
40 YB, ZB, RIA, X, YS, ZS, DELX )
41 DT(J) = SUM(1)
42 DTA(J) = SUM(2)
43 IF (JZ.FQ.NBEA(1,LB)) GO TO 20
44 GO TO 30
45 CONTINUE
46 JZ = 0
47 LR = LB+1
48 LS = LS+1
49 AR = ARH(LB)
50 NZYLR = NBEA(2,LB)
51 CONTINUE
52 RETURN
53 END

```

```

SUBROUTINE DZPY(KB,KS,LS,IZ,I,J1,J2,NYFLAG,FLND,FLNE,SGR,CGP,REFC,
1 FMACH,KR,ARB,NREA,DT,DTA,DELX )
2 YR,ZB,RIA,X,YS,ZS,DELX )
3 DIMENSION YH(1),ZR(1),RIA(1),X(1),YS(1),ZS(1),DELX(1)
4 *** GENERATES ROWS OF THE SUBMATRICES DZP, DZZ AND DZY
5 USING SUBROUTINE SUBB
6 REAL KR,M
7 COMPLEX SUM(2),DT(1),DTA(1)
8 DIMENSION NREA(2,1),ARB(1)
9 FORMAT (1H0,6F20.8)
10 NCV = 0
11 NYEL = NYFLAG
12 IND = 0
13 NZYKR = NREA(2,KB)
14 PI = 3.1415926
15 EPS = 0.00001
16 BETA = SQRT(1.0-FMACH**2)
17 FL = REFC
18 M = FMACH
19 NRV = 0
20 JZ = 0
21 LR = 1
22 JZ = 0
23 SGR = 0.0
24 CGS = 1.0
25 NZYLR = NREA(2,LR)
26 AR = ARB(LB)
27 SL = 0.0
28 CL = 1.0
29 TL = 0.0
30 LS IS THE INDEX OF THE Y AND Z COORDINATES OF SENDING POINT J
31 LS RUNS FROM NSTRIP+1 THROUGH NSTRIP+NBZ
32 DC 30 J=J1,J2
33 JR = JB+1
34 JR IS THE BODY-ELEMENT NUMBER IN BODY LR -- JB RUNS FROM 1
35 THROUGH NTZ
36 JZ = JZ+1
37 JR RUNS FROM 1 THROUGH NBE-SUB-LB
38 CALL SUBB(KB,KS,I,J,JZ,JB,LR,LS,NDY,NYFL,FLND,FLNE,PI,EPS,
39 YR,ZB,RIA,X,YS,ZS,DELX )
40 DTA(J) = SUM(1)
41 IF (JZ.EQ.NBEA(1,LR)) GO TO 20
42 GO TO 30
43 CONTINUE
44 JZ = 0
45 LR = LR+1
46 LS = LS+1
47 AR = ARB(LB)
48 NZYLR = NREA(2,LR)
49 CONTINUE
50
51
52
53

```

DZPY 54
DZPY 55

RETURN
END

```

SUBROUTINE DZYMAT ( D, DA, NFB, NLB, NTZYS,
1  LOZDY, NTAPE, ITAP2, XP, BETA, IPRNT
2  , NAREA, NSBE, NC, NS, AO, YB, ZR, AR, XIS1, XIS2, CG, SG, YP, ZP,
   DIMENSION NREA(2, ),
   DIMENSION YB(1), ZB(1), AR(1), XIS1(1), XIS2(1), CG(1), SG(1)
   DIMENSION YP(1), ZP(1)
   CALCULATION OF DZ AND DY MATRICES
   SLENDER BODY CALCULATIONS
   *****
   WORKING ARRAY USED TO STORE A ROW OF DZ OR DY
   NUMBER OF THE FIRST BODY WITH THE ORIENTATION REQ
   *****
   NLB
   NUMBER OF THE LAST BODY WITH THE ORIENTATION
   REQUESTED
   NTZYS
   NUMBER OF Z OR Y ORIENTED SLENDER BODY ELE.
   NTAPE
   I/O UNIT NUMBER WHICH THE OUTPUT MATRIX IS TO
   BE WRITTEN ON
   XP
   X-CONTROL POINT COORDINATE OF LIFTING SURFACE
   BOXES
   BETA
   SQR( 1.0 - M**2)
   *****
   COMMON /AROCOM/ NTI, MODES
   X, NP, MSTRIP, NSMAX, NCMAX, NTOTAL, NB, MSHE, MBE
   Y, ND, NE, NBY, NBZ, NTO, NTP, NTY, NTZ
   1, NTYS, NTZS, MAXGR, MAXSTR, NSBETO, NSTRI, KR, XM, REFA, REFC
   2, REFS, FMACH, LINES
   *****
   DIMENSION D(1), DA(1), XP(1)
   REAL
   INTEGER BY, BZ, C, CI, P, S, SI, YT, ZT
   NPOT = 6
   CI = 0
   SI = 0
   BZ = 0
   RY = 0
   NBYR = NB - NBY + 1
   *****
   THIS LOOP IS FOR EACH LIFTING SURF. PANEL
   *****
   400 P = 1, NP
   NSP = NS(P)
   NCP = NC(P)

```

DZYMAT 2
DZYMAT 3
DZYMAT 4
DZYMAT 5
DZYMAT 6
DZYMAT 7
DZYMAT 8
DZYMAT 9
DZYMAT10
DZYMAT11
DZYMAT12
DZYMAT13
DZYMAT14
DZYMAT15
DZYMAT16
DZYMAT17
DZYMAT18
DZYMAT19
DZYMAT20
DZYMAT21
DZYMAT22
DZYMAT23
DZYMAT24
DZYMAT25
DZYMAT26
DZYMAT27
DZYMAT28
DZYMAT29
DZYMAT30
AROCOM 2
AROCOM 3
AROCOM 4
AROCOM 5
AROCOM 6
DZYMAT32
DZYMAT33
DZYMAT34
DZYMAT35
DZYMAT36
DZYMAT37
DZYMAT38
DZYMAT39
DZYMAT40
DZYMAT41
DZYMAT42
DZYMAT43
DZYMAT44
DZYMAT45
DZYMAT46
DZYMAT47
DZYMAT48
DZYMAT49

DZYMAL02
DZYMAL03
DZYMAL04
DZYMAL05
DZYMAL06
DZYMAL07
DZYMAL08
DZYMAL09
DZYMAL10
DZYMAL11
DZYMAL12
DZYMAL13
DZYMAL14
DZYMAL15
DZYMAL16
DZYMAL17
DZYMAL18
DZYMAL19
DZYMAL20
DZYMAL21
DZYMAL22
DZYMAL23
DZYMAL24
DZYMAL25
DZYMAL26
DZYMAL27
DZYMAL28
DZYMAL29
DZYMAL30
DZYMAL31
DZYMAL32
DZYMAL33
DZYMAL34
DZYMAL35
DZYMAL36
DZYMAL37
DZYMAL38
DZYMAL39
DZYMAL40

```

C
510 IF ( NB .LT. NFB ) GC TO 650
IF ( NTV .LE. 0 ) GO TO 650
IXP = NTP
IF ( NFB .LE. 1 ) GC TO 530
NFB M1 = NFB - 1
DC 520 I = 1, NFBM1
520 IXP = IXP + NBEA(I,I)
530 CONTINUE
SGR = -1.0
CCR = 0.0
DC 600 RY = NFB, NB
DY = YB (RY)
DZ = ZB (RY)
NBEY = NBEA(1,BY)

C
C LCP FOR EACH ELEMENT OF BODY - BY -
DC 550 YI = 1, NBEY
CI = CI + 1
IXP = IXP + 1
DX = XP( IXP )

C
CALL ROWDYZ ( NFB, NLB, DY, CI, DZ,
* 2, RFTA, IDZY, NTAP, NTAP2, SGR, CCR, IPRNT
3, NSRE, AO, YB, ZB, AR, XIS1, XIS2 )
NIZYS, BY

C
550 CONTINUE
600 CONTINUE
650 CONTINUE

C
IF ( IPRNT .NE. 0 ) WRITE ( NPOT, 1000 )
* NP, NBEZ, NFB, IDZY
* RETURN

C
1000 FORMAT (16H DZYMAT*** NP =,I5, 11H NO. ROWS -,I5,15H FIRST Y BODY
* -,I5,12H LAST BODY -,I5, 8H IDZY =,I3 )
END

```



```

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
SURPOUTINE FLLD ( X01, X02, Y0, Z0, SGR, CGR, SGS, CGS, KR, K21, )
CHAR FMACH, E L, KDIR, <DIT, K2R, K21, )
CALCULATION OF THE NUMERATOR OF A DOUBLET LINE OF FINITE
LIKE KERN, THERE ARE TWO OUTPUT COMPLEX VALUES REPRESENT-
ED BY FOUR REAL NUMBERS AND AN INPUT OPTION.
WRITTEN BY D. H. LARSON, STRUCTURAL MECHANICS MDAC 11/70
* * * * *
* * * * * X - XI1
* * * * * X - XI2
* * * * * Y - ETA
* * * * * Z - ZETA
* * * * * SIN ( GAMMA-R )
* * * * * COS ( GAMMA-R )
* * * * * SIN ( GAMMA-S )
* * * * * COS ( GAMMA-S )
* * * * * REDUCED FREQUENCY
* * * * * REFERENCE LENGTH
* * * * * MACH NUMBER
* * * * *
* * * * * OPTION FLAG USED IN TKER
* * * * * REAL PART OF K01
* * * * * IMAGINARY PART OF K01
* * * * * REAL PART OF K02
* * * * * IMAGINARY PART OF K02
* * * * *
* * * * * COMMON /KDS / IND, KK1R, KK1I, KK2R, KK2I, R1
* * * * * COMMON /DLM/ K10, K20, K1R1, K1I1, K2R1, K2I1, K10I1,
* * * * * K20I2P, E2
* * * * * REAL KR, KK1R, KK1I, KK2R, KK2I
* * * * * REAL KDIR, K01R, K01I, K02R, K02I
* * * * * REAL K10I1, K20I2P, K1R1, K1I1, K2R1, K2I1, K10I1,
* * * * * COMPLEX K01, K02, K1X1, K1Y1, K2X1, K2Y1
* * * * *
* * * * * X01 = X - XI1 AND X02 = X - XI2, DELXI = XI2 - XI1
* * * * * DELXI = X01 - X02
* * * * *
* * * * * FULL KERNEL FROM - TKER -
* * * * *
* * * * * 0.0
* * * * * 0.0
* * * * * 0.0
* * * * * F **2
* * * * * KR * DELXI / CBAR
* * * * * CBAR / 2.0
* * * * * SIN ( T1 )
* * * * * COS ( T1 )
* * * * * 1
* * * * * X01

```

```

10 CALL TKEP ( X0 , CGR , Y0 , SGS , Z0 , CGS , KR , RT1 , BR , RT2 ,
* SGR , FMACH )
*
30 GC TO ( 30 , 40 ) , I
K1X11 = CMPLX ( KK1R , KK1I )
K2X11 = CMPLX ( KK2R , KK2I )
IF ( L .EQ. 0 ) GO TO 35
KD1R = KD1R - K10T1
KD2R = KD2R - K20T2P
35 CONTINUE
C
C NOW GO CALCULATE FOR XI = X12
C
X0 = X02
I = 2
GO TO 10
C
40 K1X12 = CMPLX ( KK1R , KK1I )
K2X12 = CMPLX ( KK2R , KK2I )
IF ( L .EQ. 0 ) GC TO 50
KD1R = KD1R + K10T1
KD2R = KD2R + K20T2P
50 CONTINUE
C
TEMP1 = CMPLX ( C11 , S11 )
TEMP2 = CMPLX ( C11 , -S11 )
C
C DESIRED RESULTS ( COMPLEX )
C
KD1 = K1X11 * TEMP1 - K1X12 * TEMP2
KD2 = K2X11 * TEMP1 - K2X12 * TEMP2
C
C CONVERT TO REAL AND IMAGINARY PARTS
C
KD1R = REAL ( KD1 ) + KD1R
KD1I = AIMAG ( KD1 )
KD2R = REAL ( KD2 ) + KD2R
KD2I = AIMAG ( KD2 )
RETURN
END

```

54 FLLD
55 FLLD
56 FLLD
57 FLLD
58 FLLD
59 FLLD
60 FLLD
61 FLLD
62 FLLD
63 FLLD
64 FLLD
65 FLLD
66 FLLD
67 FLLD
68 FLLD
69 FLLD
70 FLLD
71 FLLD
72 FLLD
73 FLLD
74 FLLD
75 FLLD
76 FLLD
77 FLLD
78 FLLD
79 FLLD
80 FLLD
81 FLLD
82 FLLD
83 FLLD
84 FLLD
85 FLLD
86 FLLD
87 FLLD
88 FLLD
89 FLLD
90 FLLD
91 FLLD
92 FLLD

FMMW 54
 FMMW 55
 FMMW 56
 FMMW 57
 FMMW 58
 FMMW 59
 FMMW 60
 FMMW 61
 FMMW 62
 FMMW 63
 FMMW 64
 FMMW 65
 FMMW 66
 FMMW 67
 FMMW 68
 FMMW 69
 FMMW 70
 FMMW 71
 FMMW 72
 FMMW 73
 FMMW 74
 FMMW 75
 FMMW 76
 FMMW 77
 FMMW 78
 FMMW 79
 FMMW 80
 FMMW 81
 FMMW 82
 FMMW 83
 FMMW 84
 FMMW 85
 FMMW 86
 FMMW 87
 FMMW 88
 FMMW 89
 FMMW 90
 FMMW 91
 FMMW 92
 FMMW 93
 FMMW 94
 FMMW 95
 FMMW 96
 FMMW 97
 FMMW 98
 FMMW 99
 FMMW 100
 FMMW 101
 FMMW 102
 FMMW 103
 FMMW 104
 FMMW 105

```

C
INFL = 1 ARGUMENTS
DVB = YB ( IRB )
DZB = ZB ( IRB )
DAR = AO
DELEPS = ARB ( IRB )
DELEP2 = 1.0
C = CGS
S = -SGS
DY = YS
DZ = ZS
ITYPE = 1
IF2 = 0
K ASSIGN 100 TO IRET1
GO TO 200
100 SY = 1.0
SZ = 1.0
SG
ASSIGN 200 TO IRET1
GO TO 500
200 CONTINUE
DELEPS = 1.0
DELEP2 = -1.0
C = CGS
S = SGS
DY = -YS
DZ = ZS
ITYPE = 1
IF2 = 0
K IF SENDING ELEMENT LIES ON THE Z AXIS.. DON'T USE SYMMETRY
IF ( DY .EQ. 0.0 ) GO TO 700
ASSIGN 300 TO IRET1
GO TO 200
300 CONTINUE
SY = -1.0
SZ = 1.0
SG
ASSIGN 400 TO IRET1
GO TO 500
400 CONTINUE
CHECK GROUND EFFECTS FLAG--SKIP IF ZERO
IF ( NE .EQ. 0 ) GO TO 700
PORTION FOR COMBINATION OF SYMMETRY AND GROUND EFFECTS
ITYPE = 1
IF2 = 0
K = 3
DELEPS = NE
DELEP2 = -NE
C = CGS

```


FMMW 158
 FMMW 159
 FMMW 160
 FMMW 161
 FMMW 162
 FMMW 163
 FMMW 164
 FMMW 165
 FMMW 166
 FMMW 167
 FMMW 168
 FMMW 169
 FMMW 170
 FMMW 171
 FMMW 172
 FMMW 173
 FMMW 174
 FMMW 175
 FMMW 176
 FMMW 177
 FMMW 178
 FMMW 179
 FMMW 180
 FMMW 181
 FMMW 182
 FMMW 183
 FMMW 184
 FMMW 185
 FMMW 186
 FMMW 187
 FMMW 188
 FMMW 189
 FMMW 190
 FMMW 191
 FMMW 192
 FMMW 193
 FMMW 194
 FMMW 195
 FMMW 196
 FMMW 197
 FMMW 198
 FMMW 199
 FMMW 200
 FMMW 201
 FMMW 202
 FMMW 203
 FMMW 204
 FMMW 205
 FMMW 206
 FMMW 207
 FMMW 208
 FMMW 209

```

*      BETA2, DFZ7R, DFZZI, DFZYR, DFZYI, DFYR, DFYI
      FZRP      =      C * DFZ7R      +      S * DFZYR
      FWZI      =      C * DFZZI      +      S * DFZYI
      FWZA      =      FWZ + DELEPS * CMPLX ( FWZR, FWZI )
      FWYR      =      C * DFY7R      +      S * DFYIR
      FWYI      =      C * DFYZI      +      S * DFYII
      FWYA      =      FWY + DELEPS * CMPLX ( FWYR, FWYI )
      GO TO ( 3000, 6000 ), ITYPE
      3000 GO TO ( RETL, ( 100, 200, 300, 400, 500, 600, 800, 900,
*
C
C
C      CALCULATION LOOP FOR ASSOCIATED BODIES
      5000 IF ( NAS .LE. 0 ) GO TO 3000
      ITYPE = 1
      5100 TR = NASBIT
      CHECK TO SEE IF THE ASSOCIATED BODY IS THE RECEIVING
      BODY. 0
      IF2 ( IB = NE, IRB ) GO TO 5800
      IF IT IS DETERMINE IF THE SENDING POINT IS OUTSIDE
      OR INSIDE THE BODY.
      GO TO ( 5600, 5500, 5400, 5300 ), K
      5300 IF ( OVR = NE, 0.0 ) GO TO 5800
      IF2 ( IF2 = 1
      GO TO 5800
      IF ( DYB = NE, 0.0 ) GO TO 5800
      5400 IF ( DZR = NE, 0.0 ) GO TO 5800
      5500 IF2 = 1
      5600 CONTINUE
      ETA      =      SY * YS
      ZETA     =      SZ * ZS
      YRI      =      SY * YB(1B)
      DAPIR    =      SY * YB(1B)
      DATA    =      ARR ( IR )
      TEST     =      AVT ( (FIA-YB1)**2 + (ZETA-ZB1)**2 )
      IF (TEST .GT. SCALER * DAIB) GO TO 6000
      CALL SUBI ( DAIB, ZB1, ZB2, ZB3, ZB4, ZB5, ZB6, ZB7, ZB8, ZB9, ZB10, ZB11, ZB12, ZB13, ZB14, ZB15, ZB16, ZB17, ZB18, ZB19, ZB20, ZB21, ZB22, ZB23, ZB24, ZB25, ZB26, ZB27, ZB28, ZB29, ZB30, ZB31, ZB32, ZB33, ZB34, ZB35, ZB36, ZB37, ZB38, ZB39, ZB40, ZB41, ZB42, ZB43, ZB44, ZB45, ZB46, ZB47, ZB48, ZB49, ZB50, ZB51, ZB52, ZB53, ZB54, ZB55, ZB56, ZB57, ZB58, ZB59, ZB60, ZB61, ZB62, ZB63, ZB64, ZB65, ZB66, ZB67, ZB68, ZB69, ZB70, ZB71, ZB72, ZB73, ZB74, ZB75, ZB76, ZB77, ZB78, ZB79, ZB80, ZB81, ZB82, ZB83, ZB84, ZB85, ZB86, ZB87, ZB88, ZB89, ZB90, ZB91, ZB92, ZB93, ZB94, ZB95, ZB96, ZB97, ZB98, ZB99, ZB100 )
      *
      *      FIA, ZETA, ZB1, ZB2, ZB3, ZB4, ZB5, ZB6, ZB7, ZB8, ZB9, ZB10, ZB11, ZB12, ZB13, ZB14, ZB15, ZB16, ZB17, ZB18, ZB19, ZB20, ZB21, ZB22, ZB23, ZB24, ZB25, ZB26, ZB27, ZB28, ZB29, ZB30, ZB31, ZB32, ZB33, ZB34, ZB35, ZB36, ZB37, ZB38, ZB39, ZB40, ZB41, ZB42, ZB43, ZB44, ZB45, ZB46, ZB47, ZB48, ZB49, ZB50, ZB51, ZB52, ZB53, ZB54, ZB55, ZB56, ZB57, ZB58, ZB59, ZB60, ZB61, ZB62, ZB63, ZB64, ZB65, ZB66, ZB67, ZB68, ZB69, ZB70, ZB71, ZB72, ZB73, ZB74, ZB75, ZB76, ZB77, ZB78, ZB79, ZB80, ZB81, ZB82, ZB83, ZB84, ZB85, ZB86, ZB87, ZB88, ZB89, ZB90, ZB91, ZB92, ZB93, ZB94, ZB95, ZB96, ZB97, ZB98, ZB99, ZB100
      *
      *      DAPIR, ZB1, ZB2, ZB3, ZB4, ZB5, ZB6, ZB7, ZB8, ZB9, ZB10, ZB11, ZB12, ZB13, ZB14, ZB15, ZB16, ZB17, ZB18, ZB19, ZB20, ZB21, ZB22, ZB23, ZB24, ZB25, ZB26, ZB27, ZB28, ZB29, ZB30, ZB31, ZB32, ZB33, ZB34, ZB35, ZB36, ZB37, ZB38, ZB39, ZB40, ZB41, ZB42, ZB43, ZB44, ZB45, ZB46, ZB47, ZB48, ZB49, ZB50, ZB51, ZB52, ZB53, ZB54, ZB55, ZB56, ZB57, ZB58, ZB59, ZB60, ZB61, ZB62, ZB63, ZB64, ZB65, ZB66, ZB67, ZB68, ZB69, ZB70, ZB71, ZB72, ZB73, ZB74, ZB75, ZB76, ZB77, ZB78, ZB79, ZB80, ZB81, ZB82, ZB83, ZB84, ZB85, ZB86, ZB87, ZB88, ZB89, ZB90, ZB91, ZB92, ZB93, ZB94, ZB95, ZB96, ZB97, ZB98, ZB99, ZB100
      *
      *      IF ( IPRINT = 0 ) WRITE ( NPOI, 8020 )
      *      DAIB, DZ, S, INFL, CGS, SG, DY,
      *      IF ( IOUTFL ) 2000, 3000, 2000
      6000 I = I + 1
  
```

FMMW 210
 FMMW 211
 FMMW 212
 FMMW 213
 FMMW 214
 FMMW 215

IF (I - NAS) 5100, 5100, 3000
 7000 RETURN
 8000 FORMAT (14HOENTERING FMMW)
 8010 FORMAT (5(5E20.8/) / 2110)
 8020 FORMAT (7H SUBI../ 5E20.8/ 5E20.8/ 2E20.8, 2120)
 END

AD-A106 520

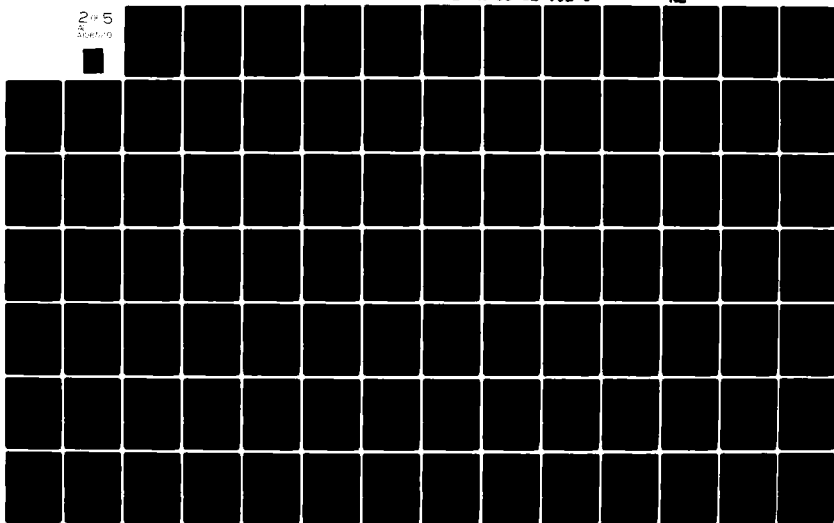
DOUGLAS AIRCRAFT CO LONG BEACH CA F/8 18/3
NUCLEAR BLAST RESPONSE COMPUTER PROGRAM. VOLUME III. PROGRAM LI--ETC(U)
AUG 81 J A MCOREW, H H CROXEN, T P KALMAN DNA001-75-C-0216

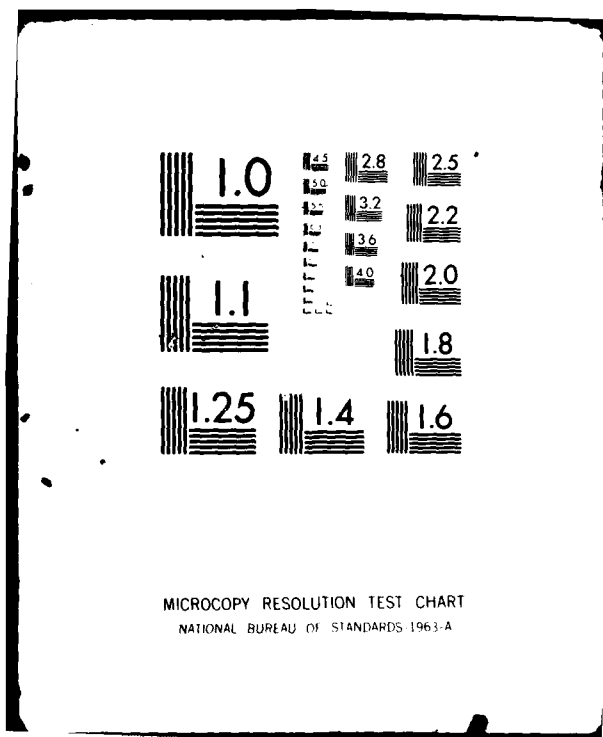
UNCLASSIFIED

AFWL-TR-81-32-VOL-3

NL

2 of 5
Page 10





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

SUBROUTINE FZY2 (X1J, X1, X2, ETA, ZETA, Y8, Z8, A, BETA2, CBAR, K,
1 FZZR, FZZI, FZYR, FZYI, FYZR, FYZI, FYVR, FYVI, MFLG)
2 *** THIS SUBROUTINE IS AN ALTERNATIVE TO SUBROUTINE FMZY ***
3 IT IS USED WHENEVER THE OPTION FLAG IBFS = 1
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

REAL M,K,KBAR,KBAR2,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,KBAR3
 DATA M,KBAR,KBAR2, LASTBR / 3*0.0, 0 /
 DATA TEST1,TEST2,CTH,STH,RAIJ,RAIJ2 / 0.142857, 0.5, 1.0, 3*0.0 /
 DATA CAPDR,CAPDI,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11 / 13*0.0 /
 10 FCRMAT (I10,10X,12HCAPDR,CAPDI /)
 20 FCRMAT (I10//10X,4HX1=F12.6,6H X1=F12.6,6H X2=F12.6/
 30 FCRMAT (I10X,5HETA=F11.6,8H ZETA=F10.6,6H Y8=F12.6,
 6H Z8=F12.6/10X,5HA=F11.6,8H BETA2=F10.6,
 8H CBAR=F12.6//)
 40 FCRMAT (I10,10X,35HM,KBAR,XA,DX,A2,RAIJ,CTH,STH /)
 50 FCRMAT (I10,10X,6HFZZR=F12.6,8H FZZI=F10.6,8H FZYI=F12.6/
 8H FZYR=F12.6,8H FYZI=F12.6,8H FYZR=F12.6,8H FVR=F12.6/
 60 FCRMAT (I10,10X,36HRWIG,RAA,CAPA,EARG,QR,QI,DELTA /)
 70 FCRMAT (I10,10X,43HI1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11 /)
 80 FCRMAT (I10,10X,3HFAU /)
 90 FCRMAT (I10,10X,8HRAI,RA2 /)
 100 FCRMAT (I10,10X,20HFTHR,FTHI,FRR,FRI /)
 IF (MFLG.NE.0) WRITE (6,30) X1J,X1,X2,ETA,ZETA,Y8,Z8,A,BETA2,CBAR
 M=SQRT(1.0-BETA2)
 IF (K.LE.0.0001 .AND. M.LE.0.0001) GO TO 110
 KRAR=2.0*K*M*A/CBAR
 KRAR2=KBAR*KBAR
 GO TO 120
 110 CONTINUE
 KBAR=0.0
 KRAR2=0.0
 120 XA=0.5*(X1+X2)
 DX=X2-X1
 A2=A*A
 FPS=0.001*A2
 IF (ETA.EQ.Y8 .AND. ZETA.EQ.Z8) GO TO 130
 PAIJ2=(ETA-Y8)*A2+(ZETA-Z8)*A2
 PAIJ=SQRT(PAIJ2)
 CTH=(ETA-Y8)/PAIJ
 STH=(ZETA-Z8)/PAIJ
 IF (PAIJ2.GT.A2) GO TO 150
 GO TO 140
 130 CONTINUE
 PAIJ=0.0
 PAIJ2=0.0
 CTH=1.0
 STH=0.0
 140 RWIG2=A2
 GO TO 160
 150 PWIG2=PAIJ2

```

160 RAA = SORT((XA -XIJ)**2 + BETA2*RWIG2)
    CT2 = CTH*CTH
    ST2 = 0.0
    IF (ABS(STH) GT 0.0001) ST2=STH*STH
    RWIG = SORT(RWIG2)
    IF (MFLG.EQ.0) GO TO 170
    WRITE (6,40)
    WRITE (6,10) M, KBAR, XA, DX, A2, RAIJ, CTH, STH
170 CONTINUE
    RAA2 = RAA * RAA
    RAA3 = RAA * RAA2
    RAA4 = RAA * RAA3
    CAPA = M - (XA -XIJ) / RAA
    DELTA = DX / RAA
    DELTA2 = DELTA * DELTA
    EARG = 0.0
    IF (KBAR LE 0.0001) GO TO 180
    EARG = KBAR * (M * (XA -XIJ) - RAA) / (BETA2 * A)
    QR = COS(EARG) / (4.0 * DX)
    QI = SIN(EARG) / (4.0 * DX)
    GO TO 190
180 QF = 1.0 / (4.0 * DX)
    QI = 0.0
190 CONTINUE
    IF (MFLG.FQ.0) GO TO 200
    WRITE (6,60)
    RWIG, RAA, CAPA, EARG, QR, QI, DELTA
200 CONTINUE
    IF (DELTA GT 0.0001) GO TO 240
    I1 = DELTA / RAA2
    TRM1 = BETA2 * A * I1
    FTHR = A * QR * TRM1
    FTHI = A * QI * TRM1
    IF (KBAR LE 0.0001) GO TO 210
    I4 = DELTA / RAA
    TRM2 = KBAR * I4
    FTHR = A * QI * TRM2
    FTHI = A * QR * TRM2
210 CONTINUE
    IF (RAIJ2 GT 0.0001) GO TO 220
    FRR = FTHR
    FRI = FTHI
    GO TO 370
220 I6 = DELTA / RAA4
    TRM1 = -3.0 * A2 * BETA2*BETA2 * I6
    CAPDR = RAIJ2 * QR * TRM1
    CAPDI = RAIJ2 * QI * TRM1
    IF (KBAR LE 0.0001) GO TO 230
    I9 = DELTA / RAA3
    TRM1 = TRM1 + KBAR2 * I1
    TRM2 = -3.0 * A * BETA2 * KBAR * I9
    CAPDR = RAIJ2 * (QR * TRM1 - QI * TRM2)

```

FZY2 54
 FZY2 55
 FZY2 56
 FZY2 57
 FZY2 58
 FZY2 59
 FZY2 60
 FZY2 61
 FZY2 62
 FZY2 63
 FZY2 64
 FZY2 65
 FZY2 66
 FZY2 67
 FZY2 68
 FZY2 69
 FZY2 70
 FZY2 71
 FZY2 72
 FZY2 73
 FZY2 74
 FZY2 75
 FZY2 76
 FZY2 77
 FZY2 78
 FZY2 79
 FZY2 80
 FZY2 81
 FZY2 82
 FZY2 83
 FZY2 84
 FZY2 85
 FZY2 86
 FZY2 87
 FZY2 88
 FZY2 89
 FZY2 90
 FZY2 91
 FZY2 92
 FZY2 93
 FZY2 94
 FZY2 95
 FZY2 96
 FZY2 97
 FZY2 98
 FZY2 99
 FZY2 100
 FZY2 101
 FZY2 102
 FZY2 103
 FZY2 104
 FZY2 105

```

230 CAPDI = RAIJ2 * (QR * TRM2 + QI * TRM1)
    FPR = FTHR + CAPDR
    GC TO 370
240 CONTINUE
    IF (DELTA - GT * TFSI2) GO TO 320
    LASTBR = 0
    TAU = (XA - XIJ) / RAA
    TAU2 = TAU * TAU
    I1 = DELTA * (1.0 - (-1.0+5.0*TAU2)*DELTA2/8.0) / RAA2
    TRM1 = A * RETA2 * I1
    FTHR = A * QR * TRM1
    FTHI = A * QI * TRM1
    IF (KBAR * LE : 0.0001) GO TO 270
    IF (LASTHR * NE : 0) GO TO 350
    DELTA3 = DELTA * DELTA2
    I2 = -(TAU * DELTA3) / (4.0 * RAA)
    I3 = DELTA3 / 12.0
    I4 = DELTA * (1.0 + (-1.0+3.0*TAU2)*DELTA2/12.0) / RAA
    I5 = -(TAU * DELTA3) / 6.0
    TRM1 = TRM1 - (KBAR2 * CAPA * I5) / (A * BETA2)
    TRM2 = KBAR * (CAPA * I2 + I4 - I3*BETA2*RWIG2/(2.0*RAA3))
    FTHR = A * (QR * TRM1 - QI * TRM2)
    FTHI = A * (QR * TRM2 + QI * TRM1)
270 IF (RAIJ2 * GT * (A2+EPS)) GO TO 280
    FPR = FTHR
    FTHI = FTHI
    GC TO 370
280 CONTINUE
    KBAR3 = KBAR * KBAR2
    IF (LASTBR * NE : 0) GO TO 340
    I6 = DELTA * (1.0 + 5.0*(-1.0+7.0*TAU2)*DELTA2/24.0) / RAA4
    TRM1 = -3.0 * A2 * BETA2 * BETA2 * I6
    CAPDR = RAIJ2 * QR * TRM1
    CAPDI = RAIJ2 * QI * TRM1
    IF (KBAR * LE : 0.0001) GO TO 310
    IF (LASTBR * NE : 0) GO TO 360
    I7 = -5.0 * TAU * DELTA3 / (12.0 * RAA3)
    I8 = DELTA3 / (12.0 * RAA2)
    I9 = DELTA * (1.0 + (-1.0+6.0*TAU2)*DELTA2/6.0) / RAA3
    I10 = -DELTA3 * TAU / (3.0 * RAA2)
    TRM1 + KBAR2 * (I1 + 3.0 * CAPA * I10)
    TRM2 + KBAR3 * CAPA * I2 / (A * BETA2)
    I11 = 3.0 * A * BETA2 * KBAR * (QR * TRM1 - QI * TRM2)
    I12 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I13 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I14 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I15 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I16 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I17 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I18 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I19 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I20 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I21 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I22 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I23 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I24 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I25 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I26 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I27 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I28 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I29 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I30 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I31 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I32 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I33 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I34 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I35 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I36 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I37 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I38 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I39 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I40 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I41 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I42 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I43 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I44 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I45 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I46 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I47 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I48 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I49 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I50 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I51 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I52 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I53 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I54 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I55 = RAIJ2 * (QR * TRM1 - QI * TRM2)
    I56 = RAIJ2 * (QR * TRM2 + QI * TRM1)
    I57 = RAIJ2 * (QR * TRM1 - QI * TRM2)

```

```

RA12 = (X1 - XIJ)**2 + BETA2 * RWIG2
RA22 = (X2 - XIJ)**2 + BETA2 * RWIG2
RA1 = SORT(RA12)
RA2 = SORT(RA22)
IF (MFLG.EQ.0) GO TO 330
WRITE (6,90)
WRITE (6,10) RA1, RA2
330 CONTINUE = ((X2-XIJ)/RA2 - (X1-XIJ)/RA1) / (BETA2*RWIG2)
GO TO 250
340 CONTINUE = RA1 * RA12
RA13 = RA2 * RA22
RA23 = ((X2-XIJ)/RA23 - (X1-XIJ)/RA13 + 2.0*I1)/(3.0*BETA2*RWIG2)
GO TO 290
350 PART1 = 0.5 * DX * (XA - XIJ)
I2 = -((PART1+RAA2)/RA2 + (PART1-RAA2)/RA1) / (BETA2*RWIG2)
DENOM = X1 - XIJ + RA1
IF (ABS(DENOM) .LE. 0.0001) STOP
I11 = ALOG(ABS((X2 - XIJ + RA2) / DENOM))
I13 = I11 - 2.0*(XA - XIJ)*I2 - RAA2 * I1
DENOM4 = SORT(BETA2) * RWIG
ARG1 = (X2 - XIJ) / DENOM4
APG2 = (X1 - XIJ) / DENOM4
I14 = (ATAN(ARG1) - ATAN(ARG2)) / DENOM4
I15 = 0.5 * ALG(RA22 / RA12) - (XA - XIJ) * I4
GO TO 260
360 CONTINUE = - (1.0/RA23 - 1.0/RA13) / 3.0 - (XA - XIJ) * I6
I7 = I1 - 2.0 * (XA - XIJ) * I7 - RAA2 * I6
I8 = ((X2-XIJ)/RA22 - (X1-XIJ)/RA12 + I4) / (2.0*BETA2*RWIG2)
I9 = -((PART1 + RAA2)/RA22 + (PART1 - RAA2)/RA12 +
1 (XA - XIJ) * I4) / (2.0 * BETA2 * RWIG2)
GO TO 300
370 CONTINUE = (RA1J2, LF, (A2-EPS) . OR . MFLG.EQ.0) GO TO 380
IF (RA1J2, LF, (A2-EPS) . OR . MFLG.EQ.0) GO TO 380
WRITE (6,20)
WRITE (6,10) CAPOR, CAPDI
380 CONTINUE = (MFLG.EQ.0) GO TO 390
IF (MFLG.EQ.0) GO TO 390
WRITE (6,70)
WRITE (6,10) I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11
WRITE (6,100) FTHR, FTHI, FRR, FRI
390 CONTINUE = CT2 * FTHR + ST2 * FRR
FZZR = CT2 * FTHI + ST2 * FRI
FZZI = CT2 * FTHP + CT2 * FRR
FYYR = ST2 * FTHI + CT2 * FRI
FYYI = ST2 * FTHI + CT2 * FRI
IF (CTH.EQ.0.0) CR, SIH, EQ, 0.0) GO TO 400
FZYP = CTH * SIH * (FRR - FTHI)
FZVI = CTH * SIH * (FRI - FTHI)

```

FZY2 158
FZY2 159
FZY2 160
FZY2 161
FZY2 162
FZY2 163
FZY2 164
FZY2 165
FZY2 166
FZY2 167
FZY2 168
FZY2 169
FZY2 170
FZY2 171
FZY2 172
FZY2 173
FZY2 174
FZY2 175
FZY2 176
FZY2 177
FZY2 178
FZY2 179
FZY2 180
FZY2 181
FZY2 182
FZY2 183
FZY2 184
FZY2 185
FZY2 186
FZY2 187
FZY2 188
FZY2 189
FZY2 190
FZY2 191
FZY2 192
FZY2 193
FZY2 194
FZY2 195
FZY2 196
FZY2 197
FZY2 198
FZY2 199
FZY2 200
FZY2 201
FZY2 202
FZY2 203
FZY2 204
FZY2 205
FZY2 206
FZY2 207
FZY2 208
FZY2 209

```

FZY2 210
FZY2 211
FZY2 212
FZY2 213
FZY2 214
FZY2 215
FZY2 216
FZY2 217
FZY2 218

```

```

400 GO TO 410
    FZYR = 0.0
    FZYI = 0.0
410 CONTINUE
    IF (MFLG.NE.0) WRITE (6,50) FZZR, FZZI, FZYR, FZYI, FYYR, FYYI
    FYZR = FZYR
    FYZI = FZYI
    RETURN
END

```

```

SUBROUTINE GEND(NPRINT, NTAPE, NTOT, IQPT, MASTDT, DT, DPZ, DPY,
1 DTA, DPZA, DPYA,
2 , IFLA, NREA, NT12, NBARAY, NCARAY, YB, ZB, ARB, AVR, RIA, XLE
3 , XTE, TH1A, TH2A, X, CG, EE, SG, YS, ZS, XIC, YIN, ZIN, DELX
4 , XLAM, SCALER)
5
6 DIMENSION IFLA(2,1), NBEA(2,1), NT12(2,1)
7 DIMENSION NBARAY(1), NCARAY(1), YB(1), ZB(1), ARB(1), AVR(1), RIA(1)
8 DIMENSION XLE(1), XTE(1), TH1A(1), TH2A(1), X(1), CG(1), EE(1)
9 DIMENSION SG(1), YS(1), ZS(1), XIC(1), YIN(1), ZIN(1), DELX(1)
10 DIMENSION XLAM(1)
11
12 *** GENERATES THE INFLUENCE COEFFICIENT MATRIX DT USING THE
13 FOLLOWING FOUR SUBROUTINES -- DPPS, DPZY, DZPY, AND DYPZ
14
15 COMMON /AROCOM/ NT1, MODES
16 X, NP, MSTRIP, NSMAX, NCMAX, NTOTL, NB, MSBE, MBE
17 Y, ND, NE, NAY, NBZ, NIO, NIP, NIV, NTZ
18
19 1, NTYS, NTZS, MAXGR, MAXSTR, NSBETO, NSTRI, KR, XM, REFA, REFC
20
21 DIMENSION NTAPE(20)
22 COMPLEX DT(1), DPZ(1), DPY(1), DTA(1), DPZA(1), DPYA(1)
23
24 REAL
25 10 FORMAT (1H1, /)
26 20 FORMAT (1H1, /)
27 25 14, 6H WITH, 14, 10H SYMMETRIC -DT- MATRIX ** ROW NO.
28 14, 6H WITH, 14, 10H ISYMMETRIC -DT- MATRIX ** ROW NO.
29 30 14, 6H WITH, 14, 10H SYMMETRIC -DT- MATRIX ** ROW NO.
31
32 ITP1 = NTAPE(1)
33 ITP2 = NTAPE(2)
34 ITP3 = NTAPE(3)
35 ITP4 = NTAPE(4)
36 ITP8 = NTAPE(8)
37
38 REWIND ITP1
39 REWIND ITP2
40 REWIND ITP3
41 REWIND ITP4
42 REWIND ITP8
43
44 EPS = 0.00001
45 PI = 3.1415926
46 FLND = FLOAT(ND)
47 FLNE = FLOAT(NE)
48 BETA = SQRT(1.0-FMACH**2)
49 FL = REF
50
51 NPV = 0
52 NROX = 0
53 LBO = 1
54 LSO = 1
55 JBO = 1
56 KR = 0
57 KT = 0
58 DP = (0.0,0.0)
59 DC = (1,500)
60 NT(1) = (0.0,0.0)

```

```

40 CONTINUE
I1 = I
I2 = NTP
J1 = I
J2 = NTP

C DPP-LOOP = 1
C K IS THE PANEL NUMBER ASSOCIATED WITH RECEIVING POINT I
C KS IS THE STRIP NUMBER ASSOCIATED WITH RECEIVING POINT I
NBXR = NCARAY(K)
DO 60 I=I1,I2
SGR = SG(KS)
CGR = CG(KS)
CALL DPPS(K,KS,I,J1,J2,SGR,CGR,REFC,FMACH,Y,Z,S,NBARAY,
1 NCARAY,DT,DTA
2 YB,ZB,ARB,AVR,XLE,XTE,X,CG,FF,SG,XIC,DELX,XLAM, SCALER )
WRITE (I1,I2) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
IF (I.EQ.I2) GO TO 60
IF (I.EQ.NBARAY(K)) K=K+1
IF (I.EQ.NBXR) GO TO 50
GO TO 60
50 CONTINUE
KS = KS+1
NBXR = NBXR+NCARAY(K)
60 CONTINUE
NSTRIP = KS
NZYSV = 0
DO 70 J=J1,J2
DT(J) = (0.0,0.0)
70 CONTINUE
IF (NB.EQ.0) GO TO 320
IF (NTZ.EQ.0) GO TO 180
I1 = I2+1
I2 = I2+NTZ

C DPZ-LOOP ** ALSO USED FOR GENERATING THE DPY-MATRIX -- SEE
C COMMENT IN DPY-LOOP BELOW
80 CONTINUE
KB = K8+1
C KB IS THE BODY NUMBER ASSOCIATED WITH RECEIVING POINT I
I2 = 0
KT = K+1
C KT IS THE INDEX OF THE ARRAY OF FIRST-AND-LAST-ELEMENTS FOR THETA-1
FOR ALL BODIES (ARRAY NAME IS IFLA(100,2))
ICOUNT = 1
IFL = 1
NZYKB = NBEA(2,KB)
IFIRST = IFLA(1,KT)
ILAST = IFLA(2,KT)
DO 170 I=I1,I2

```

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

GEND 102
 GEND 103
 GEND 104
 GEND 105
 GEND 106
 GEND 107
 GEND 108
 GEND 109
 GEND 110
 GEND 111
 GEND 112
 GEND 113
 GEND 114
 GEND 115
 GEND 116
 GEND 117
 GEND 118
 GEND 119
 GEND 120
 GEND 121
 GEND 122
 GEND 123
 GEND 124
 GEND 125
 GEND 126
 GEND 127
 GEND 128
 GEND 129
 GEND 130
 GEND 131
 GEND 132
 GEND 133
 GEND 134
 GEND 135
 GEND 136
 GEND 137
 GEND 138
 GEND 139
 GEND 140
 GEND 141
 GEND 142
 GEND 143
 GEND 144
 GEND 145
 GEND 146
 GEND 147
 GEND 148
 GEND 149
 GEND 150
 GEND 151
 GEND 152
 GEND 153

```

DO 90 J=J1,J2
DPZ(J) = (0.0,0.0)
DPY(J) = (0.0,0.0)
DPZA(J) = (0.0,0.0)
DPYA(J) = (0.0,0.0)
90 CONTINUE
CALL DPZY(
1   KB,I2,I,J1,J2,IFIRST,ILAST,REFC,FMACH,YB,ZB,
2   AVR,ARB,THIA,TH2A,NT12,NBARAY,NCARAY,NZYKB,DPZ,DPY,
3   XLE,XTE,X,CG,EE,SG,YS,ZS,XIC,DELX,XLAM,SCALER)
GO TO (100,100,110), NZYKB
100 CONTINUE
WRITE (ITP1) (DPZ(J), J=J1,J2), (DPZA(J), J=J1,J2)
IF (NZYKB.EQ.1) GO TO 120
110 CONTINUE
WRITE (ITP4) (DPY(J), J=J1,J2), (DPYA(J), J=J1,J2)
120 CONTINUE
IF (I2.FQ.NBEA(1,KB)) GO TO 130
IF (I2.EQ.ILAST.AND.ICOUNT.LT.IFL) GO TO 160
GO TO 170
130 CONTINUE
I2 = 0
IF (NZYSV.LE.1.AND.NZYKB.GE.2) GO TO 140
GO TO 150
140 CONTINUE
LBO = KB
LSC = NSTRIIP+LBO
JBO = I-NBEA(1,KB)-NBOX+1
150 CONTINUE
NZYSV = NZYKB
IF (I.FQ.I2) GO TO 180
KB = KB+1
ICOUNT = 0
IFL = 1
NZYKB = NBEA(2,KB)
160 CONTINUE
KT = KT+1
ICOUNT = ICOUNT+1
IFIRST = IFLA(1,KT)
ILAST = IFLA(2,KT)
170 CONTINUE
180 CONTINUE
IF (I2.EQ.NTO) GO TO 190

C DPY-LOOP ** THIS LOOP IS REDUCED TO SETTING THE CORRECT INDICES
C AND USING THE DPZ-LOOP ABOVE
I1 = I2+1
I2 = NTO
GO TO 80
190 CONTINUE
I1 = 1
I2 = NTP
  
```

```

IF (INTZ.EQ.0) GO TO 250
C DZP-LOOP = 1
C K IS THE PANEL NUMBER ASSOCIATED WITH RECEIVING POINT I
C KS = 1
C KS IS THE STRIP NUMBER ASSOCIATED WITH RECEIVING POINT I
C NBXR = NCARAY(K)
C KB = 0
C HEPE KB=0 SERVES AS A FLAG INDICATING THAT THE RECEIVING POINT I
C IS ON A PANEL AND NOT ON A BODY
J1 = J2+1
J2 = J2+NTZ
DO 210 I=1,I2
LS = NSTRIP+1
SGR = SG(KS)
CGR = CG(KS)
CALL DZPY(KB,KS,LS,I2,I,J1,J2,NYFLAG,FLND,FLNE,SGR,CGR,REFC,
1 FMACH,KR,ARB,NBEA,DT,DTA,DELX )
2 YB, ZB, RIA, X, YS, ZS, DELX )
WRITE (11P2) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
IF (I.EQ.I2) GO TO 210
IF (I.EQ.NBARAY(K)) K=K+1
IF (I.EQ.NBXR) GC TC 200
GO TO 210
200 CONTINUE
KS = KS+1
NBXR = NBXR+NCARAY(K)
210 CONTINUE

NYELAG = 0 ** ALSO USED FOR GENERATING THE DZY MATRIX -- SEE
C DZZ-LOOP
C COMMENT IN DZY-LOOP BELOW
C KB = 1
C KS IS THE BODY NUMBER ASSOCIATED WITH RECEIVING POINT I
C KS = NSTRIP+1
I2 = 0
I1 = I2+1
I2 = I2+NTZ
SGR = 0.0
CGR = 1.0
220 CONTINUE
LS = NSTRIP+1
LSX = LS
DO 240 I=1,I2
LS = LSX
I2 = I2+1
C KS IS THE INDEX OF THE Y AND Z COORDINATES OF RECEIVING POINT I
C IN THE DZZ-LOOP KS RUNS FROM (NSTRIP+1) THROUGH (NSTRIP+NBZ)
C IN THE DZY-LOOP KS RUNS FROM (NSTRIP+NB-NBY+1) THROUGH (NSTRIP+NB)
C IN CALL DZPY(KR,KS,LS,I2,I,J1,J2,NYFLAG,FLND,FLNE,SGR,CGR,REFC,DTA
1 FMACH,KR,ARB,NBEA,DT,DTA

```

GEND 154
 GEND 155
 GEND 156
 GEND 157
 GEND 158
 GEND 159
 GEND 160
 GEND 161
 GEND 162
 GEND 163
 GEND 164
 GEND 165
 GEND 166
 GEND 167
 GEND 168
 GEND 169
 GEND 170
 GEND 171
 GEND 172
 GEND 173
 GEND 174
 GEND 175
 GEND 176
 GEND 177
 GEND 178
 GEND 179
 GEND 180
 GEND 181
 GEND 182
 GEND 183
 GEND 184
 GEND 185
 GEND 186
 GEND 187
 GEND 188
 GEND 189
 GEND 190
 GEND 191
 GEND 192
 GEND 193
 GEND 194
 GEND 195
 GEND 196
 GEND 197
 GEND 198
 GEND 199
 GEND 200
 GEND 201
 GEND 202
 GEND 203
 GEND 204
 GEND 205

```

2  , YB, ZR, RIA, X, YS, ZS, DELX )
WRITE (IT,P2) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
IF (IZ.FQ.NBEA(I,KB)) GO TO 230
230 CONTINUE
I7 = 0
KB = KB+1
KS = KS+1
240 CONTINUE
IF (NTY.EQ.0) GO TO 320
IF (NYFLAG.NE.0) GO TO 250

C DZY-LOOP ** THIS LOOP IS REDUCED TO SETTING THE CORRECT INDICES
C AND USING THE DZZ-LOOP ABOVE
I1 = NTC-NTY+1
I2 = NTO
NYFLAG = 1
KR = LBO
KS = LSO
SGR = -1.0
CGR = 0.0
GO TO 220

250 CONTINUE
IF (NTY.EQ.0) GO TO 320
I1 = 1
I2 = NTP

C DYP-LOOP
KS = 1
KB = 0
NBXR = NCARAY(K)
J1 = NTO-NTY+1
J2 = NTO
SGR = SG(KS)
CGR = CG(KS)
DO 270 I=I1,I2
CALL DYPZ(KR,KS,LS,IZ,I,J1,J2,NYFLAG,FLND,FLNE,SGR,CGR,REFC,
1 FMAC-I,KR,ARB,NBEA,LBO,LSO,JBO,DT,DTA
2 , YB, ZR, RIA, X, YS, ZS, DELX )
WRITE (IT,P3) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
IF (I.EQ.NBARAY(K)) K=K+1
IF (I.FQ.NBXR) GO TO 260
GO TO 270
260 CONTINUE
KS = KS+1
NBXR = NBXR+NCARAY(K)
SGR = SG(KS)
CGR = CG(KS)
270 CONTINUE
NYFLAG = 0
I7 = 0

```

206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257

```

IF (NTZ.EQ.0) GO TO 310
C DYZ-LOOP ** ALSO USED FOR GENERATING THE DYY MATRIX -- SEE
C COMMENT IN DYZ-LOOP BELOW
I1 = I2+1
I2 = I2+NTZ
KS = NSTOIP+1
KR = 1
SGR = 0.0
CGR = 1.0
280 CONTINUE I=I1,I2
DO 300 I=I1,I2
  LC = 1
  IZ = IZ+1
  CALL FMAC1(KR,KS,LS,IZ,I,J1,J2,NYFLAG,FLND,FLNE,SGR,CGR,REFC,
    1 YR, YR,RIA,X,YS,ZS,CEIX)
  2 WRITE (ITP3) (DY(J), J=J1,J2), (DTA(J), J=J1,J2)
  IF (IZ.EQ.NREA(1,KR)) GO TO 290
  GO TO 300
290 CONTINUE
  IZ = 0
  KR = KR+1
  KS = KS+1
300 CONTINUE
310 CONTINUE
  IF (NYFLAG.NE.0) GO TO 320
C DYZ-LOOP ** THIS LOOP IS REDUCED TO SETTING THE CORRECT INDICES
C AND USING THE DYZ-LOOP ABOVE
I1 = NTC-NTY+1
I2 = NTA
NYFLAG = 1
KR = LHC
KS = LSC
SGR = -1.0
CGR = 0.0
GO TO 280
320 CONTINUE
  REWIND ITP1
  REWIND ITP2
  REWIND ITP3
  REWIND ITP4
  REWIND ITP8
  I1 = 1
  I2 = NTP+NTZ
  NYFLAG = 0
  ITAPE = 1
330 CONTINUE I=I1,I2
DO 360 I=I1,I2
  J1 = 1
  J2 = NTP

```

```

      READ (ITAPE) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
      IF (NTZ.EQ.0) GO TO 340
      J1 = J2+1
      J2 = J2+NTZ
      READ (ITP2) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
340  CONTINUE
      IF (NTY.EQ.0) GO TO 350
      J1 = J2+1
      J2 = J2+NTY
      READ (ITP3) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
350  CONTINUE
      WRITE (ITP8) (DT(J), J=J1,J2), (DTA(J), J=J1,J2)
      IF (IOPT.EQ.1) WRITE (MASID) (DT(J), J=1, J2),
1    (DTA(J), J=1, J2)
      IF (NPRINT.NE.3) GO TO 360
      WRITE (6,20) I,J2
      WRITE (6,30) (DT(J), J=1, J2)
      WRITE (6,25) I,J2
      WRITE (6,30) (DTA(J), J=1, J2)
      IF ((J2/5).LT.25) GO TO 360
      WRITE (6,10)
360  CONTINUE
      IF (NTY.EQ.0) GO TO 370
      IF (NYFLAG.NE.0) GO TO 370
      NYFLAG = 1
      I1 = I2+1
      I2 = I2+NTY
      ITAPE = 4
      GO TO 330
370  CONTINUE
      RETURN
      END

```

```

GEND 310
GEND 311
GEND 312
GEND 313
GEND 314
GEND 315
GEND 316
GEND 317
GEND 318
GEND 319
GEND 320
GEND 321
GEND 322
GEND 323
GEND 324
GEND 325
GEND 326
GEND 327
GEND 328
GEND 329
GEND 330
GEND 331
GEND 332
GEND 333
GEND 334
GEND 335
GEND 336
GEND 337
GEND 338
GEND 339
GEND 340
GEND 341

```

```

SUBROUTINE GENF(NMODES, NTOT, NTP, NB, YB, NSBETO, IPRINT,
1 LINES, NTP6, NUTL1, NUTL2, NTPH, NTPH4, NTFORC,
2 NTGF, MASTCP, MASTF2, MASTF3, JKR, H, FMULT, NASYM,
3 DELA, XIS1, XIS2, NSBEA, FORCE, GF, WORK, NG)
    DIMENSION NMODES(6), YB(1)
    DIMENSION H(1), FMULT(1), DELA(1), XIS1(1), XIS2(1), NSBFA(1)
    COMPLEX FORCE(1), GF(1), WORK(1)
    DIMENSION CODE(12)
    DATA IZERO, IREAD, IWRITE, 0, 1, 0, GLCJ, CNDT / 2*1.0 /
    DATA CODE / 4HOPUS, 4HFGPS, 4HDOZOS, 4HFGZS, 4HDOYOS, 4HFGYS,
1 4HDOPOA, 4HFGPA, 4HDOZOA, 4HFGZA, 4HDOYOA, 4HFGYA /
    10 FORMAT ( 1H1 /// 10X, 22H** SEGMENT 7 *** /// 5X,
1 68HCALCULATE FORCES FOR MODES AND GUST, AND GENERALIZED
2 FORCES
30 FORMAT ( // 20H FORCE COLUMN, 13, 13H OF MATRIX ; 1A4 / )
40 FORMAT ( // 20H GUST FORCE COLUMN, 13, 13H OF MATRIX ; 1A4 / )
50 FORMAT ( // 32H GENERALIZED FORCE COLUMN, 13,
1 13H OF MATRIX, 1A4 / )
60 FORMAT ( // 32H GENERALIZED GUST FORCE COLUMN, 13,
1 13H OF MATRIX, 1A4 / )
70 FORMAT ( 6F16.6 )
72 FORMAT ( 1H1 /// )
    WRITE (NTP6, 10)
    REWIND NUTL1
    REWIND NUTL2
    REWIND NTGF
    ISYM = 1
    NMODE = NMODES(1) + NG
    IF (NB .EQ. 0) GO TO 110
    IZY = 0
    IF (NB .EQ. 0) GO TO 110
    IZY = 1
    NUTL = NUTL1
    REWIND NTPH
    CONTINUE
    NSLEND = NSBETO + NR
    READ (NTPH) NMSB, NMASB
    WRITE (NUTL) NMSB, NMASB
    NMTO = NMSB
    DO 104 L = 1, 2
    DO 102 NM = 1, NMTO
    READ (NTPH) ID
    READ (NTPH) (FMULT(I), I=1, NSLEND)
    LJ2 = 0
    IH = 0
    DO 100 LB = 1, NR
    LJ1 = LJ2 + 1
    LJ2 = LJ2 + NSBEA(LB)
    DO 90 LJ = LJ1, LJ2

```

54 GENF
 55 GENF
 56 GENF
 57 GENF
 58 GENF
 59 GENF
 60 GENF
 61 GENF
 62 GENF
 63 GENF
 64 GENF
 65 GENF
 66 GENF
 67 GENF
 68 GENF
 69 GENF
 70 GENF
 71 GENF
 72 GENF
 73 GENF
 74 GENF
 75 GENF
 76 GENF
 77 GENF
 78 GENF
 79 GENF
 80 GENF
 81 GENF
 82 GENF
 83 GENF
 84 GENF
 85 GENF
 86 GENF
 87 GENF
 88 GENF
 89 GENF
 90 GENF
 91 GENF
 92 GENF
 93 GENF
 94 GENF
 95 GENF
 96 GENF
 97 GENF
 98 GENF
 99 GENF
 100 GENF
 101 GENF
 102 GENF
 103 GENF
 104 GENF
 105 GENF

```

    = IH + 1
    H(IH) = GLCJ*(FMULT(LJ) + FMULT(LJ+1)) / 2.0
    90 CONTINUE = LJ2 + 1
    100 CONTINUE
C
    WRITE (NUTL) ID
    WRITE (NUTL) (H(IH), IH = 1, NSBETO)
    102 CONTINUE
    READ (NTPH) ID
    WRITE (NUTL) ID
    IF (NMA5B.EQ.0) GO TO 106
    NMT0 = NMA5B
    104 CONTINUE
    106 CONTINUE
    IF (IZY.FQ.2) GO TO 110
    IZY = 2
    NUTL = NUTL2
    GO TO 80
    110 CONTINUE
C
    IF (JKR.GT.1) GO TO 500
C
    WRITE THE TWO MATRICES HPS AND HPA ON TAPE NTFORC IN
    COLUMN ORDER PRECEDED BY ROW- AND COLUMN-DIMENSIONS
C
    REWIND NTPH4
    IGO = 1
    READ (NTPH4) NSYM, NASYM
    NMODE = NSYM
    330 CONTINUE
    GO 340 J = 1, NMODE
    READ (NTPH4)
    READ (NTPH4) (H(I), I = 1, NTP)
    WRITE (NTFORC) (H(I), I = 1, NTP)
    340 CONTINUE
C
    IF (NASYM.FQ.0.OR.IG0.EQ.2) GO TO 350
    IGO = 2
    NMODE = NASYM
    READ (NTPH4)
    GO TO 330
C
    350 CONTINUE
    IF (NB.EQ.0) GO TO 410
C
    WRITE THE FOUR MATRICES HZS, HZA, HYS, HYA ON TAPE
    NTFORC IN COLUMN ORDER PRECEDED BY ROW- AND COLUMN-DIMENSIONS
C
    NUTL = NUTL1
    IGO = 3
    360 NMODE = NSYM
  
```

GENF 106
GENF 107
GENF 108
GENF 109
GENF 110
GENF 111
GENF 112
GENF 113
GENF 114
GENF 115
GENF 116
GENF 117
GENF 118
GENF 119
GENF 120
GENF 121
GENF 122
GENF 123
GENF 124
GENF 125
GENF 126
GENF 127
GENF 128
GENF 129
GENF 130
GENF 131
GENF 132
GENF 133
GENF 134
GENF 135
GENF 136
GENF 137
GENF 138
GENF 139
GENF 140
GENF 141
GENF 142
GENF 143
GENF 144
GENF 145
GENF 146
GENF 147
GENF 148
GENF 149
GENF 150
GENF 151
GENF 152
GENF 153
GENF 154
GENF 155
GENF 156
GENF 157

```

370      REWIND NUTL
      READ (NUTL)
      DO 390 J = 1, NMODE
      READ (NUTL)
      READ (NUTL) (H(I), I = 1, NSBETO)
      WRITE (NTFORC) (H(I), I = 1, NSBETO)
      CONTINUE
390      IF (IGO.EQ.6) GO TO 410
      IF (NASYM.EQ.0 .OR. IGO.EQ.4) GO TO 400
      NMODE = NASYM
      IGO = IGO + 1
      GO TO 370
400      CONTINUE = NUTL2
      IGO = 5
      IF (NASYM.EQ.0) IGO = 6
      GO TO 360
410      CONTINUE
      H MATRICES ARE NOW SAVED ON TAPE NTFORC
500      CONTINUE
      MM = 0
      K2 = 0
      IGO = 1
      NMODE = NMODE(1) + NG
      NMODE = NMODE(1) + NG
      NTAPE = NTP
      NTAPS = MASTCP
      DO 130 I = 1, NROW
      FMULT(I) = DELA(I)
      CONTINUE
      LENGTH = NTOT
      MM = MM + 1
      MODE = NMODE(MM)
      IF (ISYM.EQ.1) REWIND NTAPE
      K1 = K2 + 1
      K2 = K1 + 1
      MP = 0
      IGUST = 0
      MG = 0
      NCOL = NMODE
      WRITE (NTGF) MODE, NCOL
      IBEG = -NROW + 1

```

```

C      READ INTO CORE (WORK-ARRAY) THE ENTIRE SYMMETRIC, OR
C      ANTISYMMETRIC SOLUTION MATRIX (PRESSURES OR BODY FORCES)
      DO 150 N = 1, NMODE
      IBEG = IBEG + NROW
      CALL RWREC(IREAD, NTAPE, WORK(1BEG), LENGTH, 1, 0)
150 CONTINUE
160 CONTINUE
      MP = MP + 1
      KJ = (MP-1) * NROW
      DO 170 I = 1, NROW
      KJ = KJ + 1
      FORCE(I) = FMULT(I) * WORK(KJ)
170 CONTINUE
      ONE COLUMN OF FORCES IS COMPLETE, WRITE IT ON TAPE NTFORC
      CALL RWREC(IWRITE, NTFORC, FORCE, NROW, 1, 0)
      IF (IPRINT .EQ. 0) GO TO 200
      IF (IGUST .NE. 0) GO TO 180
      WRITE (NTP6,30) MP, CODE(K1)
      GO TO 190
180 MG = MG + 1
      WRITE (NTP6,40) MG, CODE(K2)
190 CONTINUE
      WRITE (NTP6,70) (FORCE(I), I = 1, NROW)
200 CONTINUE
      BEGIN GENERALIZED FORCES
      CALL ZEROUT ( GF, NTOT, 1, 0 )
      REWIND NTPHS
      READ (NTPHS) MODEM, IDUMY
      IF (ISYM .EQ. 1) GO TO 209
      DO 204 NH = 1, MODEH
      READ (NTPHS)
      READ (NTPHS)
204 CONTINUE
      READ (NTPHS)
      READ (NTPHS)
208 CONTINUE
      DO 220 MD = 1, MODE
      READ (NTPHS) ICOL
      READ (NTPHS) (H(JH), JH=1, NROW)
      DO 210 IR = 1, NROW
      GF(MD) = GF(MD) + FORCE(IR) * H(IR)
210 CONTINUE
220 CONTINUE
      WRITE GENERALIZED FORCE COLUMN ON SCRATCH TAPE NTGF
      CALL RWREC(IWRITE, NTGF, GF, MUDE, 1, 0)

```

GENE 158
 GENE 159
 GENE 160
 GENE 161
 GENE 162
 GENE 163
 GENE 164
 GENE 165
 GENE 166
 GENE 167
 GENE 168
 GENE 169
 GENE 170
 GENE 171
 GENE 172
 GENE 173
 GENE 174
 GENE 175
 GENE 176
 GENE 177
 GENE 178
 GENE 179
 GENE 180
 GENE 181
 GENE 182
 GENE 183
 GENE 184
 GENE 185
 GENE 186
 GENE 187
 GENE 188
 GENE 189
 GENE 190
 GENE 191
 GENE 192
 GENE 193
 GENE 194
 GENE 195
 GENE 196
 GENE 197
 GENE 198
 GENE 199
 GENE 200
 GENE 201
 GENE 202
 GENE 203
 GENE 204
 GENE 205
 GENE 206
 GENE 207
 GENE 208
 GENE 209

210 GENE 210
 211 GENE 211
 212 GENE 212
 213 GENE 213
 214 GENE 214
 215 GENE 215
 216 GENE 216
 217 GENE 217
 218 GENE 218
 219 GENE 219
 220 GENE 220
 221 GENE 221
 222 GENE 222
 223 GENE 223
 224 GENE 224
 225 GENE 225
 226 GENE 226
 227 GENE 227
 228 GENE 228
 229 GENE 229
 230 GENE 230
 231 GENE 231
 232 GENE 232
 233 GENE 233
 234 GENE 234
 235 GENE 235
 236 GENE 236
 237 GENE 237
 238 GENE 238
 239 GENE 239
 240 GENE 240
 241 GENE 241
 242 GENE 242
 243 GENE 243
 244 GENE 244
 245 GENE 245
 246 GENE 246
 247 GENE 247
 248 GENE 248
 249 GENE 249
 250 GENE 250
 251 GENE 251
 252 GENE 252
 253 GENE 253
 254 GENE 254
 255 GENE 255
 256 GENE 256
 257 GENE 257
 258 GENE 258
 259 GENE 259
 260 GENE 260
 261 GENE 261

```

C
  IF (MP .EQ. (NMODE-NG)) IGUST = 1
  IF (MP .LT. NMODE) GO TO 160

  GO TO (230,250,260), IGO
  230 CONTINUE
  IF (NB .NE. 0) GO TO 232
  MM = 3
  K2 = 6
  GO TO 260
  232 CONTINUE
  NTAPE = MASTFZ
  NTPHS = NUTL1
  NROW = NSBETO
  LENGTH = NROW
  IGO = 2
  JB = 1
  JE = 1
  NSBEA(1) = 1
  DO 240 I = 1, NROW
    FMULT(I) = 1.0
  IF (ABS(YB(JB)) .LT. 0.0001) FMULT(I) = 0.5
  IF (I .LT. JB + 1) JE = JB + 1
  JB = JB + 1
  JE = JE + NSBEA(JB)
  240 CONTINUE
  GO TO 140
  250 CONTINUE
  NTAPE = MASTFY
  NTPHS = NUTL2
  IGO = 3
  GO TO 140
  260 CONTINUE
  IF (ISYM .EQ. 2 .OR. NASYM .EQ. 0) GO TO 270
  NMODE = NMODES(4) + NG
  ISYM = 2
  IGO = 1
  GO TO 122
  270 CONTINUE

  READ BACK THE GENERALIZED FORCES FROM THE SCRATCH TAPE NTGF
  INTO CURE (WORK-ARRAY). AND WRITE ENTIRE MATRIX ON THE
  MASTER TAPE NTFORC ---- ALSO, PRINT THE GF COLUMNS

  REWIND NTGF
  K2 = 0
  I1 = 1
  IGO = 1
  280 CONTINUE
  MG = 0
  
```

GENF 262
GENF 263
GENF 264
GENF 265
GENF 266
GENF 267
GENF 268
GENF 269
GENF 270
GENF 271
GENF 272
GENF 273
GENF 274
GENF 275
GENF 276
GENF 277
GENF 278
GENF 279
GENF 280
GENF 281
GENF 282
GENF 283
GENF 284
GENF 285
GENF 286
GENF 287
GENF 288
GENF 289
GENF 290
GENF 291
GENF 292
GENF 293
GENF 294
GENF 295
GENF 296
GENF 297

```

C
IGUST = 0
READ (NTGF, 1) NROW, NCOL
K1 = K2 + 1
K2 = K1 + 1

ILAST = 0
DO 310 N = 1, NCOL
  IBEG = ILAST + 1
  ILAST = ILAST + NROW
  CALL RWREC(IKREAD, NTGF, WORK(IBEG), NROW, 1, 0)
  IF (IGUST.NE.0) GO TO 290
  WRITE (NTP6, 50) N, CODE(K1)
  GO TO 300
290 MG = MG + 1
  WRITE (NTP6, 60) MG, CODE(K2)
300 CONTINUE
  WRITE (NTP6, 70) (WORK(II), II = IBEG, ILAST)
  IF (N.EQ.(NCOL-NG)) IGUST = 1
310 CONTINUE

C
CALL RWREC(IWRITE, NTFORC, WORK(II), ILAST, 1, 0)
C
IF (NB.EQ.0 .AND. NASYM.EQ.0) GO TO 320
IF (IGO.EQ.3 .AND. NASYM.EQ.0) GO TO 320
IF (IGO.EQ.6) GO TO 320
WRITE (NTP6, 72)
  IGO = IGO + 1
IF (NB.NF.0) GO TO 280
C
IF (IGO.GT.2 .OR. NASYM.EQ.0) GO TO 320
  IGO = 4
  K2 = 6
GO TO 280
C
320 RETURN
END

```



```

140 FORMAT (1H0,10X,14,24H R-I ELEMENTS FOR BODY, 13 /)
150 FORMAT (1H0,6X,315,18,F12.5,7X,13)
160 157HPANEL STRIP BOX 3/4 CHORD 1/4 CHORD X-COORDINATES, 7X,
211H BOX CHORD, 5X, 9H1/4 CHORD/ 7X,15HNO. NO., 7X,1HX, 7X,
359HINBOARD CENTER OUTBOARD DELTA-X SWEEP ANGLE /)
170 FORMAT (1H0,10X,14,24H XI-S ELEMENTS FOR BODY, 13 /)
180 FORMAT (1H0,10X,14,24H K-S ELEMENTS FOR BODY, 13 /)
190 FORMAT (1H0,10X,14,24H K-S ELEMENTS FOR BODY, 13 /)
200 FORMAT (1H0,10X,14,27H THETA ELEMENTS FOR BODY, 13 /)
202 FORMAT (1H1/// 10X, 3HXC1, 6X, 3HXC2, 10X, 3HXC3
1 10X, 3HXC4, 7X, 1HY,11X,1HZ,8X,19HDELTA-Y DEL
210 FORMAT (1H1//8X,6HSTRIP X-L.E.//)
11A-2,7X, 1HE, 10X,18HCHORD GEOMETRY ARRAYS FOR ALL BODIES ** //
220 FORMAT (1H1/// 30X, 38H** GEOMETRY ARRAYS FOR ALL BODIES ** //
* 10X, 12HINTERFERENCE / 7X, 15HBODY SEGMENT, 10X,
* 3HX-1, 10X, 3HY-1, 13X, 3HZ-1,13X,3HR-1,11X,5HGAMMA / 6X,
* 16HNUMBER NUMBER /)
230 FORMAT (1H1/// 10X, 7HSLENDER / 7X, 15HBODY SEGMENT, 9X,
* 4HX1S1,11X, 4HX1S2, 13X, 2HA0, 10X, 9HA0-PRIME / 6X,
* 16HNUMBER NUMBER /)
235 FORMAT (1H1/// 5F16.5)
250 FORMAT (1H1/// 10X,13,7F12.5)
255 FORMAT (1H1///)

C
WRITE (6,120)
PI = 3.1415926
NBTO = 0
NSTC = 0
K = 0
JS = 0
LAS = 0
NZY = 0
LP1 = 2
LP2 = 1 NP
IOLD = 0
MAXSTR = 0
MAXGR = 0
NSTRIP = 0
DO 350 LP = LP1, LP2
WRITE (6,70) LP
J = 0

** READ PANEL COORDINATES, NUMBER OF CHORDWISE BOXES AND (SPANWISE)
STRIPS, AND THE ASSOCIATED BODIES
C
C
C
READ (NTI,10) X1, X2, X3, X4
READ (NTI,10) Y1, Y2, Y3, Y4
READ (NTI,20) NC, NS, IGROUP
IF (IGROUP.EQ.0) IGROUP = 1
C
C

```

```

1 WRITE (6,90) X1,X2,Y1,Z1, X3,X4,Y2,Z2, NC,NS,
  IGRUP
  NSTRIP = NSTRIP + NS
  IF (IGRUP .GT. MAXGR) MAXGR = IGRUP
C
C
C
** STORE Y- AND Z-COORDINATE OF INBOARD EDGE OF PANEL IN THE
  ARRAYS 'YIN', AND 'ZIN', RESPECTIVELY
  YIN(LP) = Y1
  ZIN(LP) = Z1
  NAT = NB
  IF (NB.EQ.0) GO TO 320
  DO 310 NA = 1, NB
  LAS = LAS+1
  NASB(LAS) = NA
  NASB(LAS) = 'NASB'
  IS THE ARRAY OF ASSOCIATED BODIES FOR ALL PANELS
C
C
C
** 310 CONTINUE
  NAS(LP) = NAT
  IS THE ARRAY OF THE NUMBER OF ASSOCIATED BODIES PER
  PANEL -- THE MAXIMUM NUMBER OF ASSOCIATED BODIES PER PANEL IS 10
C
C
C
320 CONTINUE
  NCNS = NC*NS
  NCARRAY(LP) = NC
  NSARRAY(LP) = NS
  NBARRAY(LP) = NBTO*NCNS
  NSTO = NSTO+NS
  NCPI = NC+1
  NSPI = NS+1
C
C
C
** READ THE FRACTIONAL CHORDWISE- AND SPANWISE DIVISIONS FOR
  PANEL -- ARRAYS 'TH' AND 'TAU' RESPECTIVELY
  READ (NTI,10) (TH(I), I=1,NCPI)
  READ (NTI,10) (TAU(I), I=1,NSPI)
  WRITE (6,30) NCPI,LP
  WRITE (6,60) (TH(I), I=1,NCPI)
  WRITE (6,50) NSPI,LP
  WRITE (6,60) (TAU(I), I=1,NSPI)
C
C
C
** COMPUTE DIHEDRAL ANGLE FOR PANEL -- ARRAY NAME IS 'GMA'
  YDIF = Y2-Y1
  ZDIF = Z2-Z1
  CALL ATAN3(ZDIF,YDIF,ATAN3)
  GMA(LP) = ATAN3*180.0/PI
  FOUR = X4-X3-X2+X1
  DIF31 = X3-X1
  DIF21 = X2-X1
  DIF43 = X4-X3
  DU 340 J=1,NS
  JS = JS+1
  JPI = J+1

```

GEOM 102
 GEOM 103
 GEOM 104
 GEOM 105
 GEOM 106
 GEOM 107
 GEOM 108
 GEOM 109
 GEOM 110
 GEOM 111
 GEOM 112
 GEOM 113
 GEOM 114
 GEOM 115
 GEOM 116
 GEOM 117
 GEOM 118
 GEOM 119
 GEOM 120
 GEOM 121
 GEOM 122
 GEOM 123
 GEOM 124
 GEOM 125
 GEOM 126
 GEOM 127
 GEOM 128
 GEOM 129
 GEOM 130
 GEOM 131
 GEOM 132
 GEOM 133
 GEOM 134
 GEOM 135
 GEOM 136
 GEOM 137
 GEOM 138
 GEOM 139
 GEOM 140
 GEOM 141
 GEOM 142
 GEOM 143
 GEOM 144
 GEOM 145
 GEOM 146
 GEOM 147
 GEOM 148
 GEOM 149
 GEOM 150
 GEOM 151
 GEOM 152
 GEOM 153

```

154 GEOM 154
155 GEOM 155
156 GEOM 156
157 GEOM 157
158 GEOM 158
159 GEOM 159
160 GEOM 160
161 GEOM 161
162 GEOM 162
163 GEOM 163
164 GEOM 164
165 GEOM 165
166 GEOM 166
167 GEOM 167
168 GEOM 168
169 GEOM 169
170 GEOM 170
171 GEOM 171
172 GEOM 172
173 GEOM 173
174 GEOM 174
175 GEOM 175
176 GEOM 176
177 GEOM 177
178 GEOM 178
179 GEOM 179
180 GEOM 180
181 GEOM 181
182 GEOM 182
183 GEOM 183
184 GEOM 184
185 GEOM 185
186 GEOM 186
187 GEOM 187
188 GEOM 188
189 GEOM 189
190 GEOM 190
191 GEOM 191
192 GEOM 192
193 GEOM 193
194 GEOM 194
195 GEOM 195
196 GEOM 196
197 GEOM 197
198 GEOM 198
199 GEOM 199
200 GEOM 200
201 GEOM 201
202 GEOM 202
203 GEOM 203
204 GEOM 204
205 GEOM 205

FACT = 0.5*(TAU(J)+TAU(JP1))
TDIF = TAU(JP1) - TAU(J)
DO 330 I=1,NC
  IP1 = I+1
  K = K+1
  TH13 = 0.25*TH(I) + 0.75*TH(IP1)
  TH31 = 0.75*TH(I) + 0.25*TH(IP1)

** COMPUTE THE 3/4- AND 1/4-CHORD X-COORDINATES OF ALL PANEL
BOXES * ALSO, THE AVERAGE BOX-CHORD -- STORE THEM IN THE
ARRAYS 'X', 'X1', 'X12', AND 'DELX' RESPECTIVELY
C
X(K) = FACT * (TH13*FOUR+DIF31) + TH13*DIF21 + X1
X11(K) = TAU(J) * (TH31*FOUR+DIF31) + TH31*DIF21 + X1
X12(K) = TAU(JP1) * (TH31*FOUR+DIF31) + TH31*DIF21 + X1
DELX(K) = FACT * (TH(IP1)-TH(I))*FOUR + (TH(IP1)-TH(I))*DIF21
DELA(K) = TDIF * DELX(K) * Sqrt(YDIF**2 + ZDIF**2)
C
XC(1,K) = TAU(J) * (TH(I) * FOUR+DIF31) + TH(I) * DIF21 + X1
XC(2,K) = TAU(J) * (TH(IP1)*FOUR+DIF31) + TH(IP1) * DIF21 + X1
XC(3,K) = TAU(JP1) * (TH(I) * FOUR+DIF31) + TH(I) * DIF21 + X1
XC(4,K) = TAU(JP1) * (TH(IP1)*FOUR+DIF31) + TH(IP1) * DIF21 + X1
C
IF (ABS(YDIF) .GT. 0.0001) GO TO 330
IF (ABS(Y1) .LE. 0.0001) DELA(K) = 0.5 * DELA(K)
330 CONTINUE
C
** COMPUTE DATA ARRAYS FOR ALL STRIPS OF PANEL --
YS = Y-COORDINATE OF CENTERLINE OF STRIP
ZS = Z-COORDINATE OF CENTERLINE OF STRIP
DYS = DELTA-Y OF STRIP
DZS = DELTA-Z OF STRIP
EE = SEMI-WIDTH OF STRIP
CS = CHORDLENGTH OF STRIP
SG = SINE OF THE DIHEDRAL ANGLE OF STRIP
CG = COSINE OF THE DIHEDRAL ANGLE OF STRIP
X1J = X-COORDINATE OF LEADING EDGE OF STRIP
GMAR = DIHEDRAL ANGLE OF STRIP IN RADIANS
C
YS(JS) = FACT * YDIF + Y1
ZS(JS) = FACT * ZDIF + Z1
DYS(JS) = TDIF * YDIF
DZS(JS) = TDIF * ZDIF
EE(JS) = 0.5*SQRT( DYS(JS)**2 + DZS(JS)**2)
SG(JS) = FACT * FOUR + DIF21
CG(JS) = DZS(JS)/(2.0*EE(JS))
X1J(JS) = DYS(JS)/(2.0*EE(JS))
GMAR(JS) = FACT * DIF31 + X1
COORD(JS) = YS(JS)
IF (ABS(CG(JS)) .LE. 0.0001) COORD(JS) = ZS(JS)
340 CONTINUE

```



```

C
390 LPAGE = LPAGE + 1
    WRITE (6,160)
    CONTINUE
392 DO 392 I=1, NBOX
    WRITE (6,250) I, (XC(J,I), J = 1, 4)
    CONTINUE
    LPAGE = LPAGE + 1
    DO 400 J=1, NSTRI
    WRITE (6,250) J, YS(J), ZS(J), DYS(J), DZS(J), EE(J), CS(J), XI(J)
    JL = J
    COORD(JL) = COORD(J)
    IF (LPAGE * LINES - NE - J) GO TO 400
    LPAGE = LPAGE + 1
    WRITE (6,255)
    WRITE (6,210)
    CONTINUE
400 I2 = NBOX
    J = NSTRI
    NTO = NBOX
    KI = 0
    IS2 = 0
    LBN = 0
    NBZ = 0
    NBY = 0
    NTZ = 0
    NTY = 0
    NTZS = 0
    NTYS = 0
    NSBETO = 0
    IT12 = 0
    ** PANEL GEOMETRY COMPUTATIONS COMPLETE -- BYPASS BODY DATA-READ
    AND ARRAY FORMATIONS IF NB=0
    IF (NB.EQ.0) GO TO 560
    DO 550 LR=1,NR
    ** READ BODY DATA
    READ (NT1,10) ZC, YC, RAD, AR
    READ (NT1,20) NBE, NSHE, NRI, NRS, NTL
    WRITE (6,100) LB, YC, ZC, RAD, AR, NBE, NSBE, NZY, NRI, NRS
    ** NZY=1 MEANS BODY IS Z-ORIENTED
    ** NZY=2 MEANS BODY IS BOTH Z- AND Y-ORIENTED
    ** NZY=3 MEANS BODY IS Y-ORIENTED
    ** NOTE THAT BODY DATA HAS TO BE INPUT IN THIS ORDER, I.E.
    Z-ORIENTED BODIES FIRST, Z&Y ORIENTED BODIES NEXT, AND Y-ORIENTED
    BODIES LAST
    NTO = NTO + NBE
    NSBETO = NSBETO + NSBE

```

```

410      GO TO (410,410,420), NZY
      CONTINUE
      NBZ = NBZ+1
      NTZ = NTZ + NBE
      NTZS = NTZS + NSBE
      IF (NZY.EQ.1) GO TO 430
420      CONTINUE
      NBY = NBY+1
      NTY = NTY + NBE
      NTYS = NTYS + NSBE
430      CONTINUE
      LBN = LBN+1

      **      SAVE BODY ARRAYS
      ZB(10) = ZC
      YB(10) = YC
      AVR(10) = AR
      ARB(10) = AR
      NBEA(1,10) = NBE
      NBEA(2,10) = NZY
      NSBEA(10) = NSBE
      NBEPI = NBE+1

      ZR( LBN) = ZC
      YR( LBN) = YC
      AVR( LBN) = AR
      ARB( LBN) = AR
      NBEA(1,LBN) = NBE
      NBEA(2,LBN) = NZY
      NSBEA(LBN) = NSBE
      NBEPI = NBE+1

      **      IF NBE = 0 (NO INTERFERENCE BODY ELEMENTS), BYPASS READING OF
      ALL INTERFERENCE BODY ARRAYS
      IF (NBE.EQ.0) GO TO 47C

      **      READ X-COORDINATES OF INTERFERENCE BODY ELEMENT ENDPOINTS
      READ (NTI,10) (XII(I), I=1,NBEPI)
      WRITE (6,130) NBEPI, LB
      WRITE (6,60) (XII(I), I=1,NBEPI)
      IF (NRI.NE.0) GO TO 450
      DO 440 I=1,NBEPI
      RI(I) = RAD
      CONTINUE
      GO TO 460
440      CONTINUE
450      **      READ RADII OF INTERFERENCE BODY ELEMENT ENDPOINTS -- ONLY
      IF NRI = 1
      READ (NTI,10) (RI(I), I=1,NBEPI)

```

GEOM 310
 GEOM 311
 GEOM 312
 GEOM 313
 GEOM 314
 GEOM 315
 GEOM 316
 GEOM 317
 GEOM 318
 GEOM 319
 GEOM 320
 GEOM 321
 GEOM 322
 GEOM 323
 GEOM 324
 GEOM 325
 GEOM 326
 GEOM 327
 GEOM 328
 GEOM 329
 GEOM 330
 GEOM 331
 GEOM 332
 GEOM 333
 GEOM 334
 GEOM 335
 GEOM 336
 GEOM 337
 GEOM 338
 GEOM 339
 GEOM 340
 GEOM 341
 GEOM 342
 GEOM 343
 GEOM 344
 GEOM 345
 GEOM 346
 GEOM 347
 GEOM 348
 GEOM 349
 GEOM 350
 GEOM 351
 GEOM 352
 GEOM 353
 GEOM 354
 GEOM 355
 GEOM 356
 GEOM 357
 GEOM 358
 GEOM 359
 GEOM 360
 GEOM 361

```

460 CONTINUE
WRITE (6,140) NBEPI, LB
WRITE (6,60) (RI(I), I=1,NBEPI)
** READ THETA-1-MU = ANGULAR ORIENTATION OF POINT MU ON BODY
C -- INPUT THETA-1-MU IN DEGREES --
C -- SAVE CONTINUOUS ARRAY FOR ALL BODIES IN THIA(100) IN RADIANS
C
READ (NT1,10) (TH1(I), I=1,NT1)
WRITE (6,200) NT1, LB
WRITE (6,60) (TH1(I), I=1,NT1)
IA = 0
IT11 = IT12+1
IT12 = IT12+NT1
DO 520 IT=IT11,IT12
IA = IA+1
THIA(IT) = TH1(IA)*PI/180.0
520 CONTINUE
470 CONTINUE
NSBEPI = NSBE + 1
** IF NSBE=0 (NO SLENDER BODY ELEMENTS), BYPASS READING OF
C ALL SLENDER BODY ARRAYS
C
IF (NSBE.EQ.0) GO TO 510
** READ X-COORDINATES OF SLENDER BODY ELEMENT ENPOINTS
C
READ (NT1,10) (XIS(I), I=1,NSBEPI)
WRITE (6,170) NSBEPI, LB
WRITE (6,60) (XIS(I), I=1,NSBEPI)
IF (NRS.NE.0) GO TO 490
DO 480 I=1,NSBEPI
RS(I) = RAD
480 CONTINUE
GO TO 500
490 CONTINUE
** READ RADII OF SLENDER BODY ELEMENT ENPOINTS -- ONLY
C IF NRS = 1
C
READ (NT1,10) (RS(I), I=1,NSBEPI)
500 CONTINUE
WRITE (6,180) NSBEPI, LB
510 CONTINUE
WRITE (6,60) (RS(I), I=1,NSBEPI)
C
IFLA(1,LB) = I
IFLA(2,LB) = NBE
NT12(1,LBN) = NT1
NT12(2,LBN) = NT2
I1 = I2+1
I2 = I2+NBE
K = 0

```

GEOM 362
 GEOM 363
 GEOM 364
 GEOM 365
 GEOM 366
 GEOM 367
 GEOM 368
 GEOM 369
 GEOM 370
 GEOM 371
 GEOM 372
 GEOM 373
 GEOM 374
 GEOM 375
 GEOM 376
 GEOM 377
 GEOM 378
 GEOM 379
 GEOM 380
 GEOM 381
 GEOM 382
 GEOM 383
 GEOM 384
 GEOM 385
 GEOM 386
 GEOM 387
 GEOM 388
 GEOM 389
 GEOM 390
 GEOM 391
 GEOM 392
 GEOM 393
 GEOM 394
 GEOM 395
 GEOM 396
 GEOM 397
 GEOM 398
 GEOM 399
 GEOM 400
 GEOM 401
 GEOM 402
 GEOM 403
 GEOM 404
 GEOM 405
 GEOM 406
 GEOM 407
 GEOM 408
 GEOM 409
 GEOM 410
 GEOM 411
 GEOM 412
 GEOM 413

GEOM 466
 GEOM 467
 GEOM 468
 GEOM 469
 GEOM 470
 GEOM 471
 GEOM 472
 GEOM 473
 GEOM 474
 GEOM 475
 GEOM 476
 GEOM 477
 GEOM 478
 GEOM 479
 GEOM 480
 GEOM 481
 GEOM 482
 GEOM 483
 GEOM 484
 GEOM 485
 GEOM 486
 GEOM 487
 GEOM 488
 GEOM 489
 GEOM 490
 GEOM 491
 GEOM 492
 GEOM 493
 GEOM 494
 GEOM 495
 GEOM 496
 GEOM 497
 GEOM 498
 GEOM 499
 GEOM 500
 GEOM 501
 GEOM 502
 GEOM 503
 GEOM 504
 GEOM 505
 GEOM 506
 GEOM 507
 GEOM 508
 GEOM 509
 GEOM 510
 GEOM 511
 GEOM 512
 GEOM 513
 GEOM 514
 GEOM 515
 GEOM 516
 GEOM 517

```

550 CONTINUE
560 CONTINUE
C
  IHD(8) = NSBEIO
  IHD(9) = NSTRIP
  IHD(10) = MAXSTR
  CALL WRAERO (4HHEAD,O, IHD,NOUT,NER)
  DO 561 I=1,NBOX
    TEMP(I) = DELA(I)
    TEMP(I+NBOX) = DELX(I)
    TEMP(I+2*NBOX) = XIC(I)
561  N = 3*NBOX
    DO 562 I=1,NSTRIP
      N = N+1
      TEMP(N) = CG(I)
      TEMP(N+NSTRIP) = CS(I)
      TEMP(N+2*NSTRIP) = EE(I)
      TEMP(N+3*NSTRIP) = SG(I)
      TEMP(N+4*NSTRIP) = YS(I)
      TEMP(N+5*NSTRIP) = ZS(I)
      TEMP(N+6*NSTRIP) = XIJ(I)
562  ITEMP(N+7*NSTRIP) = ISSR(I)
      N = N+7*NSTRIP
    DO 563 I=1,MAXSTR
      N = N+1
      TEMP(N) = COORD(I)
563  DO 564 I=1,NP
      N = N+1
      ITEMP(N) = NBARAY(I)
      ITEMP(N+NP) = NCARAY(I)
564  N = N+NP
      DO 565 I=1,NB
      N = N+1
      ITEMP(N) = NSBEA(I)
      ITEMP(N+NB) = YB(I)
      TEMP(N+2*NB) = ZB(I)
565  CALL WRAERO (4HGEOM,O,TEMP,NOUT,NER)
      DO 567 I=1,NBOX
      TEMP(I) = XI1(I)
      TEMP(I+NBOX) = ETA1(I)
      TEMP(I+2*NBOX) = ZETA1(I)
      TEMP(I+3*NBOX) = XI2(I)
      TEMP(I+4*NBOX) = ETA2(I)
      TEMP(I+5*NBOX) = ZETA2(I)
567  CALL WRAERO (4HGEOT,O,TEMP,NOUT,NER)
      DO 568 I=1,NBOX
      TEMP(I) = X(I)
      TEMP(I+NBOX) = ETA(I)
      TEMP(I+2*NBOX) = ZETA(I)
568  CALL WRAERO (4HGEOT,O,TEMP,NOUT,NER)
C
    IF (NB .EQ. 0) GO TO 610
  
```

```

C
DO 569 I=1, NSBEIO
TEMP(I) = XIS1(I)
TEMP(I+1*NSBEIO) = ETAS(I)
TEMP(I+2*NSBEIO) = ZETAS(I)
TEMP(I+3*NSBEIO) = XIS2(I)
TEMP(I+4*NSBEIO) = ETAS(I)
TEMP(I+5*NSBEIO) = ZETAS(I)
569 CALL WRAERO (4HGEO5,0,TEMP,NOU,T,NER)

C
JB = NSTRIP
IB = NBARAY(NP)
WRITE (6,220)
LPAGE = 1
DO 580 LB = 1, NB
JB = JB + 1
I2 = NBEA(1, LB)
DO 570 I = 1, I2
IB = IB + 1
KB = IB - NBOX
WRITE (6,235) LB, KB, X(1B), YS(JB), ZS(JB), RIA(KB), GMAR(JB)
IF (LPAGE*LBINES - NE, KB) GO TO 570
LPAGE = LPAGE + 1
WRITE (6,220)
570 CONTINUE
580 CONTINUE

C
WRITE (6,230)
LPAGE = 1
IB = 0
DO 600 LB = 1, NB
I2 = NSBEA(LB)
DO 590 I = 1, I2
IB = IB + 1
WRITE (6,235) LB, IB, XIS1(IB), XIS2(IB), AO(IB), AOP(IB)
IF (LPAGE*LBINES - NE, IB) GO TO 590
LPAGE = LPAGE + 1
WRITE (6,230)
590 CONTINUE
600 CONTINUE

C
590 CONTINUE
610 RETURN
END

```

```

GEOM 518
GEOM 519
GEOM 520
GEOM 521
GEOM 522
GEOM 523
GEOM 524
GEOM 525
GEOM 526
GEOM 527
GEOM 528
GEOM 529
GEOM 530
GEOM 531
GEOM 532
GEOM 533
GEOM 534
GEOM 535
GEOM 536
GEOM 537
GEOM 538
GEOM 539
GEOM 540
GEOM 541
GEOM 542
GEOM 543
GEOM 544
GEOM 545
GEOM 546
GEOM 547
GEOM 548
GEOM 549
GEOM 550
GEOM 551
GEOM 552
GEOM 553
GEOM 554
GEOM 555
GEOM 556
GEOM 557
GEOM 558
GEOM 559
GEOM 560
GEOM 561

```



```

SUBROUTINE GUST( NMT, NWTA, KR, CBAR, X, YS, ZS, SG, NB, NBARAY, NCARAY,
1 NBEA, FMACH, NOUT, WGS, WGA, NDM, NMT, NASYM, COL,
2 IPRINT, LINES, NOUT, WGS, WGA, NDM, NMT, NASYM, COL,
3 COMPLEX WGR, WGL, WGS(350), WGA(350), COL(350)
4 DIMENSION NBARAY(1), NCARAY(1), NBEA(2,1),
5 X(1), YS(1), ZS(1), SG(1), CG(1), CR(3,1)
6 REAL
7 KR
8
9 NTZ = (NTOT - NBOX) / 2
10 DO 80 NR = 1, NG
11 R = FMACH / (1.0 + FMACH * CR(1, NR))
12 CONST = 2.0 * KR * R / CBAR
13 LS = 1
14 LP = 1
15 NBXS = NCARAY(LP)
16
17 CALL ZEROUT(WGS, NTOT, 1, 0)
18 CALL ZEROUT(WGA, NTOT, 1, 0)
19 II = 0
20 DO 30 I = 1, NBOX
21 SING = SG(LS)
22 COSG = CG(LS)
23
24 II = II + 1
25 THR = -SING * CR(2, NR) + COSG * CR(3, NR)
26 THL = SING * CR(2, NR) + COSG * CR(3, NR)
27 XL = X(II) * CR(1, NR) + ZS(LS) * CR(3, NR)
28 YTRM = YS(LS) * CR(2, NR)
29 XLR = XL + YTRM
30 XLL = XL - YTRM
31 ARGR = CONST * XLR
32 ARGJ = CONST * XLL
33 WGR = THR * CMPLX(COS(ARGR), -SIN(ARGR))
34 WGL = THL * CMPLX(COS(ARGL), -SIN(ARGL))
35
36 WGS(II) = 0.5 * (WGR + WGL)
37 WGA(II) = 0.5 * (WGR - WGL)
38
39 IF (I.LT.NBXS) GO TO 30
40 LS = LS + 1
41
42 IF (I.EQ.NBARAY(LP)) LP = LP + 1
43 NBXS = NBXS + NCARAY(LP)
44
45 30 CONTINUE
46
47 IF (NB.EQ.0) GO TO 39
48 NRM1 = NR - 1
49 NPOS = NRM1 + 2 * NMT + 2
50 IF (NASYM.EQ.0) NPOS = NPOS - 1
51 REWIND NDW
52
53

```

54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

DO 32 NX = 1, NPOS
32 READ (NDW)
C
33 READ (NDW) (COL(I), I = 1, NTOT)
C
DO 34 I = 1, NTOT
WGS(I) = WGS(I) - COL(I)
34 CONTINUE
IF (NASYM.EQ. 0) GO TO 39
IF (NG.EQ.1) GO TO 37
NGM1 = NG-1
DO 36 NX = 1, NGM1
36 READ (NDW)
37 CONTINUE
C
READ (NDW) (COL(I), I = 1, NTOT)
DO 38 I = 1, NTOT
WGA(I) = WGA(I) - COL(I)
38 CONTINUE
C
39 CONTINUE
C
WRITE (NWT) (WGS(I), I = 1, NTOT)
IF (NASYM.NE. 0)
1WRITE (NMTA) (WGA(I), I = 1, NTOT)
IF (IPRINT.EQ. 0) GO TO 70
IG2 = 0
LINE3 = LINES*3
LPAGE = NTOT / LINE3 + 1
DO 40 KG = 1, LPAGE
IG1 = IG2 + 1
IG2 = IG2 + LINE3
IF (IG2.GT. NTOT) IG2 = NTOT
WRITE (NOUT, 50) NR, (I, WGS(I), I = IG1, IG2)
IF (NASYM.NE. 0)
1WRITE (NOUT, 60) NR, (I, WGA(I), I = IG1, IG2)
C
40 CONTINUE
50 FORMAT ( 1H1///8H COLUMN, 15,15H OF SYMM. GUST //
1
60 1 FORMAT ( 3 (16, 2E13.5) )
1
70 1 FORMAT ( 3 (16, 2E13.5) )
70 CONTINUE
C
80 CONTINUE
C
RETURN
END

```

```

1 SUBROUTINE HATS( XA, YA, XO, YO, LMBDA, X14C, X34C, Y,
2 H14C, H34C, HP, IHAT, IFPRLL, NPAIR, I112, NTH,
3 SCOE, THB
4
5     THIS SUBROUTINE COMPUTES ELEMENTS OF THE H14C-, H34C- AND
6     HP- MATRIX COLUMNS FOR BOXES IN ONE SECTION OF A SUPERPANEL
7
8     DIMENSION X14C(1), X34C(1), Y(1), H14C(1), H34C(1), HP(1),
9     I112(2,1), SCOE(1), THB(1)
10    REAL
11    LMBDA
12
13    SINL = SIN(LMBDA)
14    COSL = COS(LMBDA)
15    IF (IFPRLL .EQ. 1) GO TO 10
16    XDIF = XA - XO
17    YDIF = YO
18    RHONOT = SQR(XDIF**2 + YDIF**2)
19    CALL ATAN3( YDIF, XDIF, THNOT )
20
21    ROCOS = RHONOT * COS(THNOT + LMBDA)
22    WRITE (6,110) RHONOT, THNOT, LMBDA, ROCOS, XA, YA
23    10 CONTINUE
24
25    DO 140 IPR = 1, NPAIR
26    I1 = I112(1, IPR)
27    I2 = I112(2, IPR)
28
29    DO 130 I = I1, I2
30    IF (IFPRLL .EQ. 1) GO TO 20
31    X1MXO = X14C(I) - XO
32    X3MXO = X34C(I) - XO
33    YMYO = Y(I) - YO
34    CALL ATAN3( YMYO, X1MXO, TH1 )
35    CALL ATAN3( YMYO, X3MXO, TH3 )
36
37    ROSIN1 = RHONOT * SIN(THNOT - TH1)
38    ROSIN3 = RHONOT * SIN(THNOT - TH3)
39    COST1L = COS(TH1 + LMBDA)
40    COST3L = COS(TH3 + LMBDA)
41    THA1 = -ROSIN1/ COST1L
42    THA3 = -ROSIN3/ COST3L
43    GO TO 30
44
45    20 CONTINUE
46    YTRM = Y(I) * COSL
47    THA1 = YTRM + (X14C(I) - XA) * SINL
48    THA3 = YTRM + (X34C(I) - XA) * SINL
49
50    30 CONTINUE
51
52    GO TO (40,50,70), IHAT
53    40 CONTINUE = 1.0
54    50
55    70
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

54 HATS
 55 HATS
 56 HATS
 57 HATS
 58 HATS
 59 HATS
 60 HATS
 61 HATS
 62 HATS
 63 HATS
 64 HATS
 65 HATS
 66 HATS
 67 HATS
 68 HATS
 69 HATS
 70 HATS
 71 HATS
 72 HATS
 73 HATS
 74 HATS
 75 HATS
 76 HATS
 77 HATS
 78 HATS
 79 HATS
 80 HATS
 81 HATS
 82 HATS
 83 HATS
 84 HATS
 85 HATS
 86 HATS
 87 HATS
 88 HATS
 89 HATS
 90 HATS
 91 HATS
 92 HATS
 93 HATS
 94 HATS
 95 HATS
 96 HATS
 97 HATS
 98 HATS
 99 HATS
 100 HATS
 101 HATS
 102 HATS
 103 HATS
 104 HATS
 105 HATS

```

    A3 = 1.0
    B = 0.0
    C = 0.0
    GO TO 90
50 CONTINUE = -COSL
C
60 CONTINUE
  IF (IFPRLL .EQ. 0) GO TO 72
  YIYASL = Y(I)
  YIYASL - (X14C(I) - XA) * COSL
  A3 = YIYASL - (X34C(I) - XA) * COSL
  C
  GO TO 90
C
70 CONTINUE = -SINL
  IF (IFPRLL .EQ. 1) GO TO 80
72 CONTINUE = ROCOS / COSIIL
  RHOB3 = ROCOS / COSIIL
  YIYOSQ = (Y(I) - Y0)**2
  RHOB3 = SQRT((X14C(I) - X0)**2 + YIYOSQ)
  SINIH1 = SIN(THI1)
  SINIH3 = SIN(THI3)
  A1 = (RHOB1 - RHOB3) * COSIIL
  A3 = (RHOB3 - (RHOB3 - RHOB3) * SINIH3) * RHOB3 / RHOB3
  C
  IF (IHAT .EQ. 2) GO TO 90
  TANTIL = TAN(THI1 + LMBDA)
  TANT3L = TAN(THI3 + LMBDA)
  A1 = A1 * TANTIL
  A3 = A3 * TANT3L
  C
  GO TO 90
80 A1 = 0.0
  A3 = 0.0
  C
  GO TO 90
90 CONTINUE
  WRITE (6,120) I, X14C(I), X34C(I), Y(I)
  CC WRITE (6,110) THA1, THA3, RHOB1, RHOB3, RHOB3, RHOB3
  CC WRITE (6,110) A1, A3, B, C
  C
  SUM1 = 0.0
  SUM3 = 0.0
  SUMP = 0.0
  C
  DO 100 IT = 1, NTH
  IDFI = THA1 - THB(IT)
  IDF3 = THA3 - THB(IT)
  IDFI SQ = IDFI**2
  C
  
```

```

TDF3SQ = TDF3**2
ALTRM1 = 0.0
IF (TDF1SQ .GT. .0001) ALTRM1 = ALOG(TDF1SQ)
ALTRM3 = 0.0
IF (TDF3SQ .GT. .0001) ALTRM3 = ALOG(TDF3SQ)
XMULT1 = TDF1SQ * ALTRM1
XMULT3 = TDF3SQ * ALTRM3
XMULTP = TDF3 * ( ALTRM3 + 1.0 )

IP2 = IT + 2
SUM1 = SUM1 + XMULT1 * SCOEFF(IP2)
SUM3 = SUM3 + XMULT3 * SCOEFF(IP2)
SUMP = SUMP + XMULTP * SCOEFF(IP2)

C 100 CONTINUE
CMBR1 = SCOEFF(1) + SCOEFF(2) * THA1 + SUM1
CMBR3 = SCOEFF(1) + SCOEFF(2) * THA3 + SUM3
SLOPE = SCOEFF(2) + 2.0 * SUMP
CC 110 WRITE (6,110) CMBR1, CMBR3, SLOPE
120 FORMAT ( / , 6E15.6 )
C

H14C(1) = CMBR1 * A1
H34C(1) = CMBR3 * A3
HP(1) = CMBR3 * B + SLOPE * C
C 130 CONTINUE
C 140 CONTINUE
RETURN
END

```

```

HATS 106
HATS 107
HATS 108
HATS 109
HATS 110
HATS 111
HATS 112
HATS 113
HATS 114
HATS 115
HATS 116
HATS 117
HATS 118
HATS 119
HATS 120
HATS 121
HATS 122
HATS 123
HATS 124
HATS 125
HATS 126
HATS 127
HATS 128
HATS 129
HATS 130
HATS 131
HATS 132
HATS 133
HATS 134
HATS 135
HATS 136

```

```

SUBROUTINE IDFL (EE,E2, ETA01,ZET01,ARE,AIM,BRE,BIM,CRE,CIM,
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
*** INTEGRATES THE PLANAR PARTS OF THE INCREMENTAL
OSCILLATORY KERNELS FOR UNSTEADY CASES
10 FORMAT (1H0,6E20.6/)
PI = 3.1415926
PARN = ETA01**2 - ZET01**2
FACR = PARN*ARE + ETA01*BRE + CRE
FACI = PARN*AIM + ETA01*BIM + CIM
PARNR = BRE/2.0 + ETA01*ARE
PARNI = BIM/2.0 + ETA01*AIM
UP = (ETA01-EE)**2 + ZET01**2
DOWN = (ETA01+EE)**2 + ZET01**2
ALARG2 = UP/DOWN
ALARG2 = ALOG(ALARG2)
TRM2R = PARNR * ALARG2
TRM2I = PARNI * ALARG2
TRM3R = 2.0*EE*ARE
TRM3I = 2.0*EE*AIM
AZET = ABS(ZET01)
IF (AZET/EE) .LE. 0.001) GO TO 100
TESTO = ABS((RISQX-E2)/(2.0*EE*AZET))
IF (TESTO.LE.0.0001) GO TO 110
COEF = (2.0*EE)/(RISQX-E2)
ARGA = COEF*ZET01
TEST = ABS(ARGA)
IF (TEST.LE.0.3) GO TO 120
ARGT = COEF*AZET
ATANA = ATAN(ARGT)
FUNCT = ATANA/AZET
GO TO 170
100 CONTINUE
FUNCT = (2.0*EE)/(ETA01**2-E2)
GO TO 170
110 CONTINUE
FUNCT = 0.0
GO TO 170
120 CONTINUE
ARGA**2
SER = 1./3.*S*(-1./5.*S*(1./7.*S*(-1./9.*S*(1./11.-S/13.))))
ALPHA = E2*(COEF**2)*SER
FUNCT = COEF*(1.0-ALPHA*(ZET01**2)/E2)
170 CONTINUE
FACR * FUNCT
TRMIR = FACI * FUNCT
XIIJR = TRMIR + TRM2R + TRM3R
XIIJI = TRMI + TRM2I + TRM3I
RETURN
END

```

```

SUBROUTINE IOF2(EE,E2, ETA01,ZET01,A2R,A21,B2R,B21,C2R,C21,
1  RISQX,D11JR,D11JI)
** INTEGRATES THE NONPLANAR PARTS OF THE INCREMENTAL
C OSCILLATORY KERNELS FOR UNSTEADY CASES
C IO FORMAT (I1H0, 6E20.6/)
EPS = 0.0001
AZET = ABS(ZET01)
DEN0 = RISQX-E2 + ZET01**2
PARN = ETA01**2 + ETA01*B2R + C2R
FAC1 = PARN*A21 + ETA01*B21 + C21
ETA02=ETA01**2
ZET02=ZET01**2
IF ((AZET/EE) .LE. 0.001) GO TO 120
TEST0=ABS((RISQX-E2)/(2.0*EE*AZET))
IF (TEST0.GT.0.1) GO TO 120
DEN2 = (ETA01+EE)**2*ZET02
DEN3 = (ETA01-EE)**2*ZET02
FAC2A= RISQX*ETA01+(ETA02-ZET02)*EE
FAC3A= RISQX*ETA01-EE
FAC2B= RISQX+ETA01*EE
FAC3B= RISQX-ETA01*EE
TRM2R= (FAC2A*A2R+FAC2B*B2R+(ETA01+EE)*C2R)/DEN2
TRM2I= (FAC2A*A21+FAC2B*B21+(ETA01+EE)*C21)/DEN2
TRM3R= -(FAC3A*A2R+FAC3B*B2R+(ETA01-EE)*C2R)/DEN3
TRM3I= -(FAC3A*A21+FAC3B*B21+(ETA01-EE)*C21)/DEN3
IF (TEST0.LE.0.0001) GO TO 110
COEF = (2.0*EE)/(RISQX-E2)
ARGA = COEF*ZET01
TEST = ABS(ARGA)
IF (TEST.GT.0.3) GO TO 90
SER = 1.0/3.0+S*(-1./5.0+S*(1.0/7.0+S*(-1.0/9.0+S*(1.0/11.0-S/13.0))))
ALPHA= E2*(COEF**2)*SER
FUNCT= COEF*(1.0-ALPHA*(ZET01**2)/E2)
GO TO 100
90 CONTINUE
ARGT = COEF*AZET
ATANA= ATAN(ARGT)
FUNCT= ATANA/AZET
100 CONTINUE
TRMIR= FACR*FUNCT
TRMII= FACI*FUNCT
D11JR= (TRMIR + TRM2R + TRM3R)/(2.0*ZET02)
D11JI= (TRMII + TRM2I + TRM3I)/(2.0*ZET02)
GO TO 170
110 CONTINUE
FUNCT= 0.0
GO TO 100
120 CONTINUE
DENB = (ETA01+EE)**2 + ZET01**2
DENB = (ETA01-EE)**2 + ZET01**2

```

2 IOF2
 3 IOF2
 4 IOF2
 5 IOF2
 6 IOF2
 7 IOF2
 8 IOF2
 9 IOF2
 10 IOF2
 11 IOF2
 12 IOF2
 13 IOF2
 14 IOF2
 15 IOF2
 16 IOF2
 17 IOF2
 18 IOF2
 19 IOF2
 20 IOF2
 21 IOF2
 22 IOF2
 23 IOF2
 24 IOF2
 25 IOF2
 26 IOF2
 27 IOF2
 28 IOF2
 29 IOF2
 30 IOF2
 31 IOF2
 32 IOF2
 33 IOF2
 34 IOF2
 35 IOF2
 36 IOF2
 37 IOF2
 38 IOF2
 39 IOF2
 40 IOF2
 41 IOF2
 42 IOF2
 43 IOF2
 44 IOF2
 45 IOF2
 46 IOF2
 47 IOF2
 48 IOF2
 49 IOF2
 50 IOF2
 51 IOF2
 52 IOF2
 53 IOF2

54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85

IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2
IDF2

```

UP1R = 2.0*(E2*A2R + C2R)
UP1I = 2.0*(E2*A2I + C2I)
UP2R = 4.0*E2*ETA01*B2R
UP2I = 4.0*E2*ETA01*B2I
TRM1R = (UP1R*(R1SQX+E2) + UP2R)/(DENA*DENB)
TRM1I = (UP1I*(R1SQX+E2) + UP2I)/(DENA*DENB)
IF ((AZET/EE) - LE - 0.001) GO TO 130
COEF = (2.0*EE)/(R1SQX-E2)
ARGA = COEF*ZET01
TEST = ABS(ARGA)
IF (TEST.GT.0.3) GO TO 125
S = ARGA**2
SER = 1./3.*S*(-1./5.+S*(1./7.+S*(-1./9.+S*(1./11.-S/13.))))
ALPHA = E2*(COEF**2)*SER
FUNCT = COEF*(1.0-ALPHA*(ZFT01**2)/F2)
GO TO 140
125 CONTINUE
ARGT = COEF*AZET
ATANA = ATAN(ARGT)
FUNCT = ATANA/AZET
ALPHA = (E2/ZET02)*(1.0-FUNCT*(DENO/(2.0*EE)))
GO TO 140
130 CONTINUE
ALPHA = ((2.0*E2)/(ETA02-E2))**2
140 CONTINUE
TRM2R = -ALPHA*FACR/E2
TRM2I = -ALPHA*FACI/E2
DIIJR = EE*(TRM1R + TRM2R)/DENO
DIIJI = EE*(TRM1I + TRM2I)/DENO
170 CONTINUE
RETURN
END

```


INCRD 54
INCRD 55
INCRD 56
INCRD 57
INCRD 58
INCRD 59
INCRD 60
INCRD 61
INCRD 62
INCRD 63
INCRD 64
INCRD 65
INCRD 66
INCRD 67
INCRD 68
INCRD 69
INCRD 70
INCRD 71
INCRD 72
INCRD 73
INCRD 74
INCRD 75
INCRD 76
INCRD 77
INCRD 78
INCRD 79
INCRD 80
INCRD 81
INCRD 82
INCRD 83
INCRD 84
INCRD 85
INCRD 86
INCRD 87
INCRD 88
INCRD 89
INCRD 90
INCRD 91
INCRD 92
INCRD 93
INCRD 94
INCRD 95
INCRD 96
INCRD 97
INCRD 98
INCRD 99
INCRD 100
INCRD 101
INCRD 102
INCRD 103
INCRD 104
INCRD 105

```

IND = 1
M = SQRT(1.0 - BETA**2)
BR = FL/2.
EPS = 0.00001
PI = 3.14159265
XDELX = SDELX
XDELY = DELY
EE = 0.5*XDELY
EZ = EE**2
DELR = 0.0
DELI = 0.0
TIS = 0.0
T2S = 0.0
ATIS = 0.0
AT2S = 0.0
T1 = 0.0
T2 = 0.0
COUNT = 0.
XO = AX
YO = AY
ZO = AZ
80 CONTINUE
      CALL TKER (XO,YO,ZO,KK,BR,SGR,CGR,SGS,CGS,T1,T2,M)
      AT1 = ABS(T1)
      AT2 = ABS(T2)
      IF (AT1.GT.ATIS) ATIS=AT1
      IF (AT2.GT.AT2S) AT2S=AT2
      IF (COUNT)130,90,150
      DKRC = KIRI1 - K10T1
      XRC = K2RI2P-K20T2P
      XKIC = K2IT2P
      AT2 = ABS(T2)
      JO = 1
      IF (IMG.NE.0) JO = 2 + NUBI
      IF (NDBLE.NE.0) JO = 2
      IF (IMG.NE.0.AND.NDBLE.NE.0) JO = 8 + NUBI
      IF (IR.LE.NCPN8) GO TO 110
      IF (IR.GT.NBXS) GO TO 110
      IF (LHS.NE.0) GO TO 100
      DKRI = USE1(JO,I0)
      DKII = USE2(JO,I0)
      XKRI = XUSE1(JO,I0)
      XKII = XUSE2(JO,I0)
      XO = AX2
      YO = AY2
      ZO = AZ2
      CALL TKER (XO,YO,ZO,KK,BR,SGR,CGR,SGS,CGS,T1,T2,M)
      AT1 = ABS(T1)
      AT2 = ABS(T2)
      IF (AT1.GT.ATIS) ATIS=AT1
      IF (AT2.GT.AT2S) AT2S=AT2

```

INCR0106
 INCR0107
 INCR0108
 INCR0109
 INCR0110
 INCR0111
 INCR0112
 INCR0113
 INCR0114
 INCR0115
 INCR0116
 INCR0117
 INCR0118
 INCR0119
 INCR0120
 INCR0121
 INCR0122
 INCR0123
 INCR0124
 INCR0125
 INCR0126
 INCR0127
 INCR0128
 INCR0129
 INCR0130
 INCR0131
 INCR0132
 INCR0133
 INCR0134
 INCR0135
 INCR0136
 INCR0137
 INCR0138
 INCR0139
 INCR0140
 INCR0141
 INCR0142
 INCR0143
 INCR0144
 INCR0145
 INCR0146
 INCR0147
 INCR0148
 INCR0149
 INCR0150
 INCR0151
 INCR0152
 INCR0153
 INCR0154
 INCR0155
 INCR0156
 INCR0157

DKRO = KIRTI - K10TI
 DKIO = K1ITI
 XKRO = K2RT2P-K20T2P
 XKIO = K2IT2P
 GO TO 170
 100 X0 = AXI
 Y0 = AYI
 Z0 = AZI
 CALL TKER (X0,Y0,Z0,KR,BR,SGR,CGR,SGS,CGS,T1,T2,M)
 AT1 = ABS(T1)
 AT2 = ABS(T2)
 IF (AT1.GT.ATIS) ATIS=AT1
 IF (AT2.GT.ATIS) ATIS=AT2
 DKRI = KIRTI - K10TI
 DKII = K1ITI
 XKRI = K2RT2P-K20T2P
 XKII = K2IT2P
 DKRO = USE3(JO,IO)
 DKIO = USE4(JO,IO)
 XKRC = XUSE3(JO,IO)
 XKID = XUSE4(JO,IO)
 GO TO 160
 110 CONTINUE
 COUNT = -1.
 120 X0 = AXI
 Y0 = AYI
 Z0 = AZI
 GO TO 80
 130 DKRI = KIRTI - K10TI
 DKII = K1ITI
 XKRI = K2RT2P-K20T2P
 XKII = K2IT2P
 COUNT = 1.
 X0=AX2
 Y0=AY2
 Z0=AZ2
 GO TO 80
 140 DKRO = KIRTI - K10TI
 DKIO = K1ITI
 XKRO = K2RT2P-K20T2P
 XKIC = K2IT2P
 JO = 1
 IF (IMG.NE.O) JO = 2 + NOBI
 IF (NDRLE.NE.O) JO = 2
 IF (IMG.NE.O.AND.NDRLE.NE.O) JO = 8 + NOBI
 IF (LHS.EQ.O) GO TO 170
 150 USE3(JO,IO) = DKRI
 USE4(JO,IO) = DKII
 XUSE3(JO,IO) = XKRI
 XUSE4(JO,IO) = XKII
 GO TO 180
 160 USE1(JO,IO) = DKRO
 170

```

180 USE2(J0,I0) = DKIO
XUSE1(J0,I0) = XKRO
XUSE2(J0,I0) = XKIO
CONTINUE
X0 = AX
Y0 = AY
Z0 = AZ
XIIJR = 0.0
XIIJI = 0.0
DIIJR = 0.0
DIIJI = 0.0
PI = 3.1415926
XMULT = ((XDELX)/(8.0*PI))
IF ((Y0.EQ.ZERO).AND.(Z0.EQ.ZERO)) GO TO 220
IF ((Z0.EQ.ZERO).AND.(SGS.EQ.ZERO)) GO TO 230
190 ETA01 = Y0*SGS + Z0*SGS
ZET01 = -Y0*SGS + Z0*CGS
AZETO = ABS(ZET01)
IF (AZETO.LE.0.0001) ZET01 = 0.
200 RISQX = ETA01**2 + ZET01**2
210 ARE = (DKRI - 2.*DKRC + DKRO)/(2.0*E2)
AIM = (DKII - 2.*DKIC + DKIO)/(2.0*E2)
BIM = (DKRI - DKRO)/(2.0*EE)
CRE = (DKIO - DKII)/(2.0*EE)
CIM = DKRC
GO TO 250
220 ETA01 = 0.0
ZET01 = 0.0
RISQX = 0.0
GO TO 210
230 ETA01 = Y0*CGS
ZET01 = 0.
240 RISQX = ETA01**2
GO TO 210
250 CONTINUE
IF (ATLS.EQ.0.0) GO TO 255
CALL (EE,E2,ETA01,ZET01,ARE,AIM,BRE,BIM,CRE,CIM,
1 IDF1 (EE,E2, RISQX,XIIJR,XIIJI)
DELX = XMULT*XIIJR
DELI = XMULT*XIIJI
CONTINUE
255 IF (AT2S.EQ.0.0) GO TO 260
A2R = (XKRI - 2.0*XKRC + XKRO)/(2.0*E2)
A2I = (XKII - 2.0*XKIC + XKIO)/(2.0*E2)
B2R = (XKRI - XKRO)/(2.0*EE)
B2I = (XKIO - XKII)/(2.0*EE)
C2R = XKRC
C2I = XKIC
CALL
1 IDF2(EE,E2, ETA01,ZET01,A2R,A2I,B2R,B2I,C2R,C2I,
RISQX,DIIJR,DIIJI)

```

INCR0158
 INCR0159
 INCR0160
 INCR0161
 INCR0162
 INCR0163
 INCR0164
 INCR0165
 INCR0166
 INCR0167
 INCR0168
 INCR0169
 INCR0170
 INCR0171
 INCR0172
 INCR0173
 INCR0174
 INCR0175
 INCR0176
 INCR0177
 INCR0178
 INCR0179
 INCR0180
 INCR0181
 INCR0182
 INCR0183
 INCR0184
 INCR0185
 INCR0186
 INCR0187
 INCR0188
 INCR0189
 INCR0190
 INCR0191
 INCR0192
 INCR0193
 INCR0194
 INCR0195
 INCR0196
 INCR0197
 INCR0198
 INCR0199
 INCR0200
 INCR0201
 INCR0202
 INCR0203
 INCR0204
 INCR0205
 INCR0206
 INCR0207
 INCR0208
 INCR0209

DELR = DELR + XMULT*DIJJR
 DELI = DELI + XMULT*DIJJJ
 260 CONTINUE
 C ***
 C ***
 C 265 RETURN
 END

INCR0210
 INCR0211
 INCR0212
 INCR0213
 INCR0214
 INCR0215
 INCR0216

```

C      SUBROUTINE INPUTA (NTI, NMS, NDOF, MODES, IPR
1      , ELXI, ELYI, ELZI, PHI )
C
C      DIMENSION ELXI(1), ELYI(1), ELZI(1)
C      DIMENSION PHI(NDOF,NMS,MODES)
C
C      COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINES
C      DIMENSION DEFL(8)
C      DATA DEFL/3H F,3H L,3H H,5H THETA,5H ALPHA,4H PSI,5H BETA,5H DELTA/
C      DATA AERO/4HAERO/
C
C      10 FORMAT (6F12.0)
C      20 FORMAT (1H0,20X,*MODESHAPE FOR MODE*,14,4X5HFREQ=,E11.4
1      / 5H SECT,6X,A5,8(9XA5) )
C      21 FORMAT (15,9E14.6)
C
C      REWIND NTI
C      READ (NTI)
C      READ (NTI)
C      CALL HEADNG
C      WRITE (NOUT,50)
C      50 FORMAT (1H0,4H NODE,8X7HELXI(1),8X7HELYI(1),8X7HELZI(1)/)
C      40 FORMAT (1X,14,3E15.6)
C      DO 100 I=1,NMS
C      100 READ (NTI,10) ELXI(I), ELYI(I), ELZI(I)
C      WRITE (NOUT,40) I, ELXI(I), ELYI(I), ELZI(I)
C
C      SKIP BY FREQUENCIES
C      READ (NTI,10) (DUM,I=1,MODES)
C
C      DO 130 K=1,MODES
C      LINES = KROW
C      DO 120 J=1,NMS
C      READ (NTI,10) (PHI(I,J,K),I=1,NDOF)
C      IF (IPR.EQ.0) GO TO 120
C      IF (LINES.LT.KROW) GO TO 115
C      CALL HEADNG
C      WRITE (NOUT,20) K, WR , (DEFL(I),I=1,NDOF)
C      LINES = LINES+4
C      115 WRITE (NOUT,21) J, (PHI(I,J,K),I=1,NDOF)
C      LINES = LINES+1
C      120 CONTINUE
C      130 CONTINUE
C
C      200 READ (NTI,90) CHECK
C      90 FURMAT (A4)
C      IF (CHECK.NE.AERO) GO TO 200
C
C      READ (NTI)
C      READ (NTI)
C      READ (NTI)

```

```

INPUT A 2
INPUT A 3
INPUT A 4
INPUT A 5
INPUT A 6
INPUT A 7
INPUT A 8
INPUT A 9
INPUT A 10
INPUT A 11
INPUT A 12
INPUT A 13
INPUT A 14
INPUT A 15
INPUT A 16
INPUT A 17
INPUT A 18
INPUT A 19
INPUT A 20
INPUT A 21
INPUT A 22
INPUT A 23
INPUT A 24
INPUT A 25
INPUT A 26
INPUT A 27
INPUT A 28
INPUT A 29
INPUT A 30
INPUT A 31
INPUT A 32
INPUT A 33
INPUT A 34
INPUT A 35
INPUT A 36
INPUT A 37
INPUT A 38
INPUT A 39
INPUT A 40
INPUT A 41
INPUT A 42
INPUT A 43
INPUT A 44
INPUT A 45
INPUT A 46
INPUT A 47
INPUT A 48
INPUT A 49
INPUT A 50
INPUT A 51
INPUT A 52
INPUT A 53

```

INPUT A54
INPUT A55

RETURN
END

```

SUBROUTINE MATMUL(NW, NPSTAP, NBFM, NEWBFZ, NEWBFY, NWORK, NTSBE,
1  NSBETO, LENGTH, NTOT, NMODE, NMODEB, IPRINT, KR,
2  DT, ROT, WORK, RWORK)
3
4  COMPLEX WORK(NWORK), DT(900)
5  DIMENSION RWORK(2,NWORK), ROT(2,900)
6  REAL
7
8  10 FORMAT (1H0///25X, I4, 44H SLENDER BODY ELEMENT FORCES ** MODF
9  1 NO., I4// 20X, 11HELEMENT NO., 12X, 2HFZ, 30X, 2HFY/)
10 FORMAT (1H1 ///)
11 REWIND NBFM
12 REWIND NEWBFZ
13 REWIND NEWBFY
14 NSBET2 = 2 * NSBETO
15 NBEG = 0
16 NBEG1 = NBEG + 1
17 NBEG2 = NTOT + 1
18 IFLAG = 1
19 IBEG = 2*LENGTH+1
20 DO 30 IM=1,NWORK
21 DO 40 IM = 1, NSBET2
22 CALL RWREC(IFLAG, NBFM, WORK(IBEG), LENGTH, 1, 0)
23 IBEG = IBEG + LENGTH
24
25 40 CONTINUE = 0
26 DO 90 J=1,NMODE
27 IF (IPRINT .GE. 2) WRITE (6,10) NSBETO, J
28 IFLAG = 1
29 CALL ZEROUT(DT, LENGTH, 1, 0)
30
31 *** READ ONE COLUMN OF SOLUTIONS FROM TAPE NW
32
33 CALL RWREC(IFLAG, NW, DT(1), NTOT, 1, 0)
34
35 *** APPEND THE ARRAY CONTAINING ONE COLUMN OF SOLUTIONS BY THE
36 CORRESPONDING COLUMN OF CP-Z-DELTA-A AND CP-Y-DELTA-A **
37 THESE ARE READ FROM TAPE NPSTAP
38
39 CALL RWREC(IFLAG, NPSTAP, DT(NBEG2), NTSBE, 1, 0)
40 KK = 2*LENGTH
41 DO 50 IM = 1, NSBET2
42 WORK(IM) = 0.0
43 DO 70 I = 1, NSBET2
44 NBW = NBEG + I
45 DO 60 K=1, LENGTH
46 KK = KK + 1
47
48 C
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

      RWORK(1,NBW) = RWORK(1,NBW) + RWORK(1,KK) * RDT(1,K)
      IF (KR .EQ. 0.0) GO TO 60
      RWORK(2,NBW) = RWORK(2,NBW) + RWORK(2,KK) * RDT(1,K)
      CONTINUE
60 CONTINUE
70 IFLAG = 0
      WRITE (NEWBFZ) (WORK(IWK), IWK = 1, NSBF12, 2)
      WRITE (NEWBFY) (WORK(IWK), IWK = 2, NSBF12, 2)
      IF (IPRINT .LT. 2) GO TO 90
      IMY = NBEG1 - 1
      DO 80 K = 1, NSBETO
      IMZ = IMY + 1
      IMY = IMZ + 1
      IMZ = IMZ + 1
      WRITE (6,20) K, WORK(IMZ), WORK(IMY)
      CONTINUE
80 CONTINUE
90 RETURN
      END

```

MATMUL54
 MATMUL55
 MATMUL56
 MATMUL57
 MATMUL58
 MATMUL59
 MATMUL60
 MATMUL61
 MATMUL62
 MATMUL63
 MATMUL64
 MATMUL65
 MATMUL66
 MATMUL67
 MATMUL68
 MATMUL69
 MATMUL70
 MATMUL71
 MATMUL72
 MATMUL73
 MATMUL74
 MATMUL75
 MATMUL76


```

NFSBEB = 1
NLSBEB = 0
NFU = 0
IF ( NL .LE. 0 ) GO TO 50
    THE FIRST BODY IS NOT = 1, WE MUST, THEREFORE
    DETERMINE THE LOCATION OF THE FIRST ELEMENT IN THE
    FIRST BODY.
DO 40 I = 1, NL
    NLSBEB = NLSBEB + NSBE( I, 1 )
    IH1 = IH1 + NLSBEB + 1
40 NFSBEB = NFSBEB + NSBE ( I, 1 )
    LOOP FOR EACH BODY
50 CONTINUE
IF (NR .EQ. 0) GO TO 52
R = FMACH / ( 1. + FMACH * CR(1,NR) )
CONST = TWOKR * R / CBAR
CRMULT = 0.5 * CR(3,NR)
IF (K .EQ. 3) CRMULT = 0.5 * CR(2,NR)
52 CONTINUE
DO 100 IB = NFBOOY, NLSBOY
    NLSBEB = NLSBEB + NSBE (IB, 1 )
    CALCULATE TERMS FOR EACH ELEMENT IN BODY
    OPAR = 1.0 + AR(1B)
    DEN = TWOP1 * OPAR
    IF ( K .NE. 3 ) GO TO 70
    DEN = AR( 1B )
    RAR = AR(1B)
70 CONTINUE
    NFU = NFSBEB, NLSBEB
    DO 80 IU = 1
        IT = NFSBEB, NLSBEB
        WSR = 0.0
        WST = 0.0
        WSPR = 0.0
        WSPI = 0.0
        TWDAO = 2.0 * AO(IT) * OPAR * ( XIS2(IT) - XIS1(IT) )
        D2D = 1.0 / ( DEN * AO(IT) **2 )
        ELR = TWOP1 * AOP(IT) * RAR
        ELI = TPKRC * AO(IT) * * RAR
        EM = PI * AO(IT)
        IH1 = IH1 + 1
        IH2 = IH1 + 1
        DELXB = XIS2(IT) - XIS1(IT)
        IF (NR .NE. 0) GO TO 72
        WSR = 0.5 * (DHDX(IH1) + DHDX(IH2))
        WST = ( H(IH1) + H(IH2) ) * KR / CBAR
        WSPR = (DHDX(IH2) - DHDX(IH1)) * CBAR / DELXB
        WSPI = 2.0 * KR * WSR

```

MUZYC 54
 MUZYC 55
 MUZYC 56
 MUZYC 57
 MUZYC 58
 MUZYC 59
 MUZYC 60
 MUZYC 61
 MUZYC 62
 MUZYC 63
 MUZYC 64
 MUZYC 65
 MUZYC 66
 MUZYC 67
 MUZYC 68
 MUZYC 69
 MUZYC 70
 MUZYC 71
 MUZYC 72
 MUZYC 73
 MUZYC 74
 MUZYC 75
 MUZYC 76
 MUZYC 77
 MUZYC 78
 MUZYC 79
 MUZYC 80
 MUZYC 81
 MUZYC 82
 MUZYC 83
 MUZYC 84
 MUZYC 85
 MUZYC 86
 MUZYC 87
 MUZYC 88
 MUZYC 89
 MUZYC 90
 MUZYC 91
 MUZYC 92
 MUZYC 93
 MUZYC 94
 MUZYC 95
 MUZYC 96
 MUZYC 97
 MUZYC 98
 MUZYC 99
 MUZYC100
 MUZYC101
 MUZYC102
 MUZYC103
 MUZYC104
 MUZYC105

```

C      NOTE THAT WS-PRIME = (WSPR, WSPI) IS PARTIAL WITH RESPECT
C      TO X / C8AR
C
C      UZY ( 1, IU) = WSR / D2D
C      UZY ( 2, IU) = WSI * ELR - WSI * ELI + WSPR * EM
C      CPZY ( 1, IU) = WSR * ELI + WSI * ELR + WSPR * EM
C      CPZY ( 2, IU) = WSR * ELI + WSI * ELR + WSPR * EM
C      CALCULATE DELTA-CP * AREA
C      CPZY ( 1, IU) = -CPZY ( 1, IU) * TWOAO
C      CPZY ( 2, IU) = -CPZY ( 2, IU) * TWOAO
C      GO TO 80
72 CONTINUE
C      X
C      XL = 0.5 * (XIS1(IT) + XIS2(IT))
C      YTRM = X * CR(1, NR) + ZB(18) * CR(3, NR)
C      XLR = YB(18) * CR(2, NR)
C      XLL = XL + YTRM
C      ARGRL = CONST * XLR
C      WGR = CONST * XLL
C      WGL = CMPLX( COS(ARGRL), -SIN(ARGRL) )
C      IF (K - EQ. 3) GO TO 74
C      WGS = CMPLT * (WGR + WGL)
C      WGA = CRMULT * (WGR - WGL)
C      GO TO 76
74 CONTINUE
C      WGS = CRMULT * (WGR - WGL)
C      WGA = CRMULT * (WGR + WGL)
76 CONTINUE
C
C      GRE = REAL (WGS)
C      GIM = AIMAG(WGS)
C      UZY (1, IU) = GRE / D2D
C      UZY (2, IU) = GIM / D2D
C      UZYA( IU) = WGA / D2D
C
C      PRE = ELR
C      PIM = ELI - TWOGR * R * CR(1, NR) * EM
C      CPZY (1, IU) = TWOAO * (GRE * PIM - GIM * PIM)
C      CPZY (2, IU) = TWOAO * (GRE * PIM + GIM * PIM)
C      CPZYA( IU) = TWOAO * WGA * CMPLX(PRE, PIM)
C
C      80 CONTINUE
C      INCREMENT ELEMENT NUMBER
C      IH1 = IH2
C      NFU = NFSBEB + NSBE (18, 1)
C      90 NFSBEB = NFSBEB + NSBE (18, 1)
C      100 NFSBEB = NFSBEB + NSBE (18, 1)
C      110 CONTINUE
C      RETURN
C      END
MUZYCI06
MUZYCI07
MUZYCI08
MUZYCI09
MUZYCI10
MUZYCI11
MUZYCI12
MUZYCI13
MUZYCI14
MUZYCI15
MUZYCI16
MUZYCI17
MUZYCI18
MUZYCI19
MUZYCI20
MUZYCI21
MUZYCI22
MUZYCI23
MUZYCI24
MUZYCI25
MUZYCI26
MUZYCI27
MUZYCI28
MUZYCI29
MUZYCI30
MUZYCI31
MUZYCI32
MUZYCI33
MUZYCI34
MUZYCI35
MUZYCI36
MUZYCI37
MUZYCI38
MUZYCI39
MUZYCI40
MUZYCI41
MUZYCI42
MUZYCI43
MUZYCI44
MUZYCI45
MUZYCI46
MUZYCI47
MUZYCI48
MUZYCI49
MUZYCI50
MUZYCI51
MUZYCI52
MUZYCI53
MUZYCI54
MUZYCI55
MUZYCI56
MUZYCI57

```

```

1 SUBROUTINE NACL(NRUN, NTPSH, NSBETO, NBCUM, X3L, NXQ, XIQ, NTP6)
2 NSUP, INSUP, H, DHDX, NMODE, KPRINT,
3
4 COMPUTES *H-, ALPHA-DEGREES-OF-FREEDOM* AT EACH (AERO-)
5 NACELLE NODE
6
7 DIMENSION NBCUM(1), X3L(1), XIQ(1), INSUP(1), H(1), DHDX(1)
8
9
10 40 FORMAT ( 6E15.6 )
11 50 FORMAT ( // 6H THE, 14, 37H H-ARRAY ELEMENTS FOR SUPERBODY
12 60 INO, 14, 10H, COLUMN, 14, 17H ( H D.O.F. )
13 70 INO, 14, 10H, COLUMN, 14, 40H DHDX-ARRAY ELEMENTS FOR SUPERBODY
14 80 INO, 14, 10H, COLUMN, 14, 17H ( H D.O.F. )
15 90 INO, 14, 10H, COLUMN, 14, 37H H-ARRAY ELEMENTS FOR SUPERBODY
16 100 INO, 14, 10H, COLUMN, 14, 17H ( ALPHA D.O.F. )
17 110 INO, 14, 10H, COLUMN, 14, 40H DHDX-ARRAY ELEMENTS FOR SUPERBODY
18
19 IF (NRUN.EQ. 1) REWIND NTPSH
20
21 100F
22 ICOL = 1
23 DO 190 NX = 1, NXQ
24
25 90 CONTINUE
26 ICOL = ICOL + 1
27 DO 100 I = 1, NSBETO
28 H(I) = 0.0
29 DHDX(I) = 0.0
30 CONTINUE
31 DO 160 J = 1, NSUP
32 ISB = INSUP(J)
33 IF (ISB.EQ. 1) GO TO 120
34
35 11 = NBCUM(ISB-1) + 1
36 12 = NBCUM(ISB)
37 DO 150 I = 1, 12
38 GO TO (130, 140), 100F
39 130 H(I) = 1.0
40 GO TO 150
41
42 140 DHDX(I) = -1.0
43 H(I) = -(X3L(I) - XIQ(NX))
44 CONTINUE
45
46 160 CONTINUE
47 WRITE (NTPSH) ICOL, NSBETO, ( H(I), I = 1, NSBETO)
48 WRITE (NTPSH) ICOL, NSBETO, ( DHDX(I), I = 1, NSBETO)
49
50 IF (100F.EQ. 2) GO TO 180
51 100F
52
53

```

```

IF (KPRINT .EQ. 0) GO TO 90
WRITE (NTP6,50) NSBETO, NRUN, ICOL
WRITE (NTP6,40) (H(I), I = 1, NSBETO)
WRITE (NTP6,60) NSBETO, NRUN, ICOL
WRITE (NTP6,40) (DHDX(I), I = 1, NSBETO)
GO TO 90
CONTINUE = 1
IDOF

C
IF (KPRINT .EQ. 0) GO TO 190
WRITE (NTP6,70) NSBETO, NRUN, ICOL
WRITE (NTP6,40) (H(I), I = 1, NSBETO)
WRITE (NTP6,80) NSBETO, NRUN, ICOL
WRITE (NTP6,40) (DHDX(I), I = 1, NSBETO)
CONTINUE

C
NMODE = NMODE + ICOL
RETURN
END

```

```

SUBROUTINE NEWH(NTP5, NTP6, NTAPE, MASTH4, MASTH, MASTDH, NP,
1 NTOU, NSYM, NASYM, NROW, MOJES, NBARAY,
2 NCARAY, YS, ZS, Y, X14C, X34C,
3 H14C, H34C, HP, NUDE, XI, ETA,
4 ELXI, ELYI, ELZ1, HCOL, PHI, THB,
5 IFP, IPAIR, I112, SWORK,
6 NUDETO, NDOFTO, PHIT, MDUF, MNODE)

      THIS SUBROUTINE IS THE MAIN PART OF THE *NEW H-, ALPHA-,
      THETA-SPLINE* PACKAGE

      DIMENSION NBARAY(1), NCARAY(1), YS(1), ZS(1), Y(1),
1 X14C(1), X34C(1), H14C(1), H34C(1), HP(1),
2 NODE(1), XI(1), ETA(1), ELXI(1), ELYI(1),
3 ELZI(1), HCOL(1), PHIT(NDOFTO,MOJES), IFP(1),
4 IPAIR(1), I112(2, 10, 10), WORK(NTOT, NDOFTO),
5 SWORK(1),

      DIMENSION INDX(200)
      DIMENSION PHIT(MDOF, MNODE, MOJES)
      DIMENSION DOF(5), FIX(5), IHAT(5), XOAK(20), YOAK(20), NASS(50)
      DIMENSION XAR(50), YAR(50), ZAR(50), XBR(50), YBR(50),
      REAL LMBDA
      DATA DOF, FIX, KUDE / 5*1H, 1HH, 1HA, 1HT, 1HB, 1HD, -1 /

10 FORMAT ( 6I12 )
20 FORMAT ( 5(11X, 1A1) )
30 FORMAT ( 6F12.0 )
50 FORMAT ( 1H1// )
1 LOX, 8HNSUPTO =, 14, 26H (NUMBER OF SUPERPANELS) / 10X, *** ///
2 8HNUDETO =, 14, 45H (TOTAL NUMBER OF NODES IN ALL SUPERPANELS) / 10X,
3 3NELS) / 10X, 8HNDTOFTO =, 14, 58H (TOTAL NUMBER OF DEGREES-OF-FREEDOM) / 10X,
4 4DOM IN ALL SUPERPANELS) / 10X, 8HMAXSEC =, 14, 53H (MAXIMUM NUMBER
5 OF SECTIONS IN ANY ONE SUPERPANEL) / 10X, 8HMAXPR =, 14,
6 66H (MAXIMUM NUMBER OF *FIRST-, LAST BOX PAIRS* IN ANY ONE SECT
7 ION)
60 FORMAT ( 1H1// 10X, 10HSUPERPANEL, 13, 14H INPUT VALUES // )
1 10X, 7HNSC =, 14, 22H (NUMBER OF SECTIONS) /
2 10X, 7HNODES =, 14, 22H (NUMBER OF NODES) /
3 10X, 7HDOF =, 14, 22H (DEGREES OF FREEDOM) /
4 10X, 5HXA =, 14, 22H (DEGREES OF FREEDOM) /
5 45H (INBOARD) EDGE COORD.-S OF ELASTIC AXIS) / 10X,
6 5HXB =, 14, 22H (OUTBOARD) EDGE COORD.-S OF ELASTIC AXIS) / 10X,
7 45H (OUTBOARD) EDGE COORD.-S OF ELASTIC AXIS) / 10X,
70 FORMAT ( 10X, 11HSECTION NO., 13, 15H OF SUPERPANEL, 13 //
1 10X, 8HIFPRLL =, 13
2 5HNO =, 14, 22H (NUMBER OF SECTIONS) /
3 35HNUMBER OF FIRST-, LAST BOX PAIRS =, 14, 10X,
4 23HSET FIRST-, LAST / 20X, 3HBOX, 7X, 3HBOX /
5 ( 112, 2110 )
80 FORMAT ( 10X, 31HNUODAL COORDINATES IN SUPERPANEL, 13 //
1 9X, 40HI XI(NODE) )

```

```

82 FORMAT ( // 10X, 31HASSOCIATED NODES FOR SUPERPANEL, I3,
1 24H ( TAKEN FROM SUPERPANEL, I3, 2H )// ( 8X, 12I6 ) )
90 FORMAT ( 2I10, 2F14.5 )

C
DO 94 J = 1, NODES
DO 92 I = 1, NDOFT0
PHI(I,J) = 0.0
92 CONTINUE
94 CONTINUE
KX = 0
LNTH = 200
IT = 0
CALL OPENMS(NTAPE, INDX, LNTH, IT)
LINES = 45

C
READ (NIP5,10) NSUPT0, MAXSEC, MAXPR
WRITE (NIP6,50) NSUPT0, NODETO, NDOFT0, MAXSEC, MAXPR

C
NDOF = 0
KN = 0

C
DO 270 ISP = 1, NSUPT0
READ (NIP5,10) NSEC, NODES, NEAASS, NODASS
READ (NIP5,20) DOF
READ (NIP5,30) XA, YA, ZA, XB, YB, ZB
WRITE (NIP6,60) ISP, NSEC, NODES, DOF, XA, YA, ZA, XB, YB, ZB
XAR(ISP) = XA
YAR(ISP) = YA
ZAR(ISP) = ZA
XBR(ISP) = XB
YBR(ISP) = YB
ZBR(ISP) = ZB

C
DO 100 M = 1, 5
IHAT(M) = 0
IF (DOF(M) .EQ. FIX(M)) IHAT(M) = M
100 CONTINUE

C
XDIF = XB - XA
YDIF = YB - YA
ZDIF = ZB - ZA
RHO = SQRT(YDIF**2 + ZDIF**2)
COSG = YDIF / RHO
SING = ZDIF / RHO

C
IF (NEAASS .NE. 0) GO TO 120
READ (NIP5,10) (NODE(I), I = 1, NODES)
WRITE (NIP6,80) ISP
DO 110 I = 1, NODES
ND = NODE(I)
XI(I) = ELXI(ND)
ETA(I) = (ELXI(ND) - YA) * COSG + (ELZI(ND) - ZA) * SING
120

```

NEWH 54
 NEWH 55
 NEWH 56
 NEWH 57
 NEWH 58
 NEWH 59
 NEWH 60
 NEWH 61
 NEWH 62
 NEWH 63
 NEWH 64
 NEWH 65
 NEWH 66
 NEWH 67
 NEWH 68
 NEWH 69
 NEWH 70
 NEWH 71
 NEWH 72
 NEWH 73
 NEWH 74
 NEWH 75
 NEWH 76
 NEWH 77
 NEWH 78
 NEWH 79
 NEWH 80
 NEWH 81
 NEWH 82
 NEWH 83
 NEWH 84
 NEWH 85
 NEWH 86
 NEWH 87
 NEWH 88
 NEWH 89
 NEWH 90
 NEWH 91
 NEWH 92
 NEWH 93
 NEWH 94
 NEWH 95
 NEWH 96
 NEWH 97
 NEWH 98
 NEWH 99
 NEWH 100
 NEWH 101
 NEWH 102
 NEWH 103
 NEWH 104
 NEWH 105


```

IF (ISO.EQ.0) GO TO 250
NDOF = NDOF + 1
DO 190 I = 1, NTOI
  H14C(I) = 0.0
  H34C(I) = 0.0
  HP(I) = 0.0
190 HP(I)

C
C
C      BEGIN DO LOOP ON ALL SECTIONS OF SUPERPANEL *ISP*
C
DO 200 N = 1, NSEC
  XO = XOAR(N)
  YO = YOAR(N)

  CALL HATS( XA, YA, XO, YO, LMBDA, X14C, X34C, Y
1
2
  H14C, H34C, HP, IFP(N), NPAIR, I1I2(1,1,N),
  NTH, SWORK(K),
200 CONTINUE

C
C      END DO LOOP ON *M* (1 FOR H, 2 FOR ALPHA, 3 FOR THETA)
C
KN1 = KN + 1
KN2 = KN + 2
KN3 = KN + 3
CALL WRITMSINTAPE, H14C, NTOI, KN1
CALL WRITMSINTAPE, H34C, NTOI, KN2
CALL WRITMSINTAPE, HP, NTOI, KN3

C
KN = KN3
IF (IPRINT.NE.2) GO TO 250
WRITE (NTP6,220) DOF(M), L
LCUM = LCUM + 5

C
DO 210 I = 1, NTOI
  LCUM = LCUM + 1
  WRITE (NTP6,230) I, H14C(I), H34C(I), HP(I)
  IF (LCUM.LT.LIMIT) GO TO 210
  LIMIT = LIMIT + LINES
  LCUM = LCUM + 5
  WRITE (NTP6,240) DOF(M), L
210 CONTINUE

C
220 FORMAT ( // 10X, 31HDEFLECTIONS AND SLOPES DUE TO, 1A1,
1
  17H OF NOISE NUMBER, 13 // 12X, 3HBOX, 9X, 7H1/4-C,
2
  7X, 7H-3/4-C, 8X, 6HDX/DX // )
230 FORMAT ( 115, 3F14.5 )
240 FORMAT ( 1H1 // )

C
250 CONTINUE
260 CONTINUE
270 CONTINUE

```

```

C      BEGIN MATRIX MULTIPLICATION TO OBTAIN ** AND *HP* COLUMNS
C      PER MODE (PHI-COLUMNS ARE EXPECTED TO BE *NDOFTO* LONG)
C
      DO 420 IGO = 1, 3
      GO TO (280,290,300), IGO
      280 NTHOUT = MASTH4
      GO TO 310
      290 NTHOUT = MASTH
      GO TO 310
      300 NTHOUT = MASTDH
      310 CONTINUE
C
      WRITE (NTHOUT) NSYM, NASYM
      KN = IGO - 3
C
      DO 320 J = 1, NDOF
      KN = KN + 3
      CALL READMS(NTAPE, WORK(1,J), NTOF, KN
      WRITE (NTP6,440) (WORK(I,J), I = 1, NTOF)
      320 CONTINUE
C
      M1 = 1
      M2 = NSYM
      IF (IPRINT.NE. 0) WRITE (NTP6,470)
      LOOP = 1
      LIMIT = LINES - 5
      LCUM = 0
      330 CONTINUE = 0
      DO 400 M = M1, M2
      WRITE (NTP6,470) M
      WRITE (NTP6,440) (PHI(J,M), J = 1, NDOFTO)
      MODE = MODE + 1
      DO 350 I = 1, NTOF
      HCOL(I) = 0.0
C
      DO 340 J = 1, NDOF
      HCOL(I) = HCOL(I) + WORK(I,J) * PHI(J, M)
      340 CONTINUE
      350 CONTINUE
C
      MH = M - M1 + 1
      WRITE (NTHOUT) MH
      WRITE (NTHOUT) (HCOL(I), I = 1, NTOF)
      IF (IPRINT.EQ. 0) GO TO 400
      GO TO (360,370,380), MODE
      360 WRITE (NTP6,430) MODE
      GO TO 390
      370 WRITE (NTP6,440) MODE
      GO TO 390

```

262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313

```

380 WRITE (NTP6,450) MODE
390 CONTINUE
    WRITE (NTP6,460) (HCUL(I), I = 1, NTOT)
    LCUM = LCUM + NTOT / 6
    IF (LCUM .LT. LIMIT .OR. M .EQ. M2) GO TO 400
    IF (LOOP .EQ. 1) WRITE (NTP6,470)
    LCUM = LCUM + 5
    LIMIT = LIMIT + LINES
400 CONTINUE
C
    WRITE (NTHOUT) KODE
    IF (NASYM .EQ. 0 .OR. LOOP .EQ. 2) GO TO 410
    LOOP
    M1 = NSYM + 1
    M2 = NSYM + NASYM
    IF (IPRINT .NE. 0) WRITE (NTP6,480)
    GO TO 330
C
410 CONTINUE
C
420 CONTINUE
C
430 FORMAT ( // 10X, 29HH-1/4-C MATRIX COLUMN (MODE) ; 13 /
440 FORMAT ( // 10X, 29HH-3/4-C MATRIX COLUMN (MODE) ; 13 /
450 FORMAT ( // 10X, 29HDX / DX MATRIX COLUMN (MODE) ; 13 /
460 FORMAT ( 5X, 6E15.6 )
470 FORMAT ( 1H1 // 10X, 32H*** SYMMETRIC MODES *** /
480 FORMAT ( 1H1 // 10X, 32H*** ANFISYMMETRIC MODES *** /
C
    CALL CLOSMS( NTAPE )
    REWIND MASTH4
    REWIND MASTH
    REWIND MASTDH
    RETURN
    END

```

```

NEW 314
NEW 315
NEW 316
NEW 317
NEW 318
NEW 319
NEW 320
NEW 321
NEW 322
NEW 323
NEW 324
NEW 325
NEW 326
NEW 327
NEW 328
NEW 329
NEW 330
NEW 331
NEW 332
NEW 333
NEW 334
NEW 335
NEW 336
NEW 337
NEW 338
NEW 339
NEW 340
NEW 341
NEW 342
NEW 343
NEW 344
NEW 345
NEW 346
NEW 347
NEW 348

```


54
55
56
57
58
ORGN
ORGN
ORGN
ORGN
ORGN

C 100 GO TO 7
NTAPE = NTOLD
RETURN
END

```

SUBROUTINE PISTON( NB , N3OX , NSBETO, KR , CBAR , FMACH,
1 NTPH , NTPDH , NPH4 , MASTFZ , MASTFY, NEWTPH,
2 NCARAY , NCARAY , YB , ZB , AO ,
3 XIS1 , XIS2 , CG , SG , EE , YS , ZS , H , DHDX , HP ,
4 XC , DCP , DCPS , DCPA , DFQ , DFQS , DPCA , NR, CR)
5 DCP , DCPS , DCPA , DFQ , DFQS , DPCA , NR, CR)

THIS SUBROUTINE COMPUTES PRESSURES AND BODY FORCES
DUE TO MODES AND GUSTS USING --
** PISTON THEORY **

DIMENSION NBARAY(1), NCARAY(1), NSBEA(1), YB(1), ZB(1), AO(1)
1 AOP(1), XIS1(1), XIS2(1), CG(1), SG(1), EE(1)
2 YS(1), ZS(1), XC(4,350)
3 H(1), DHDX(1), HP(1), CR(3,20)
4 DCP(1), DCPS(1), DCPA(1), DFQ(1), DFQS(1), DFQA(1)
5 DCP1, DCP2, DFQ1, DFQ2, ELAK, DCPTRM, TERM2, FPART
6 EILNOT
7 KR, KBJ, LMRNOT
8 PI / 3.1415926 /, NTP6 / 6 /
9 CODE, NTEMP / -1, 10 /

10 FORMAT ( 1H1 /// 10X, 32HENTER SUBROUTINE PISTON / )
WRITE (NTP6,10)

TWOPIOM = 2.0*PI / FMACH
FOUROM = 4.0 / FMACH
TWOM = 2.0 * FMACH
NSLEND = NSBETO + NB

REWIND MASTCP
REWIND NTEMP
IF (JRUN.EQ. 1) REWIND NEWTPH
REWIND NTPH
REWIND NTPDH
REWIND NTPH4
READ (NTPH) NSYM, NASYM
READ (NTPDH)
IF (NB.EQ. 0) GO TO 120

REWIND MASTFZ
REWIND MASTFY
NFTAPE = MASTFZ
NMT0 = NSYM
DO 110 IZY = 1, 4
DO 100 NM = 1, NMT0
READ (NTPH) ID
READ (NTPDH)
READ (NTPH) ( H (I), I = 1, NSLEND)
READ (NTPDH) ( DHDX(I), I = 1, NSLEND)
LJ2 = 0
IH = 0

```

```

C      DO 30 LB = 1, NR
      LJ1 = LJ2 + 1
      LJ2 = LJ2 + NSBEA(LB)

C      DO 20 LJ = LJ1, LJ2
      IH = IH + 1
      HB = 0.5 * (H(LJ) + H(LJ+1))
      HBP = 0.5 * (DHDX(LJ) + DHDX(LJ+1))

C      TWOAO = 2.0*AO(IH)
      AB = TWOPOD * ((AOP(IH) - 1.0/TWOM) * HB + TWOAO * HBP)
      BB = TWOPOD * TWOAO * HB / CBAR

C      DELX = XIS2(IH) - XIS1(IH)
      DFQ(IH) = -(CMPLX(AB, KR*BB)) * DELX

C      20 CONTINUE

C      LJ2 = LJ2 + 1

C      30 CONTINUE
      IF (IPRINT.LT.2) GO TO 90
      GO TO (40,50,60,70), IZY

C      40 WRITE (NTP6,440) NM
      GO TO 80

C      50 WRITE (NTP6,450) NM
      GO TO 80

C      60 WRITE (NTP6,460) NM
      GO TO 80

C      70 WRITE (NTP6,470) NM
      80 CONTINUE

C      WRITE (NTP6,480) (DFQ(I), I = 1, NSBETO)
      90 CONTINUE
      WRITE (NFTAPE) (DFQ(I), I = 1, NSBETO)

C      100 CONTINUE

C      READ THE CODE -1 (END INDICATOR OF H-, DHDX- MATRICES)
      READ (NTPH)
      READ (NTPDH)

C      NMTO = NASYM
      NFTAPE = NTEMP
      IF (IZY.NE.2) GO TO 110
      READ *HEADER* OF NEXT H-, DHDX- MATRIX
      READ (NTPH) NSYM, NASYM
      READ (NTPDH)
      NMTO = NSYM
      NFTAPE = MASTFY

C      110 CONTINUE

```

PISTON54
 PISTON55
 PISTON56
 PISTON57
 PISTON58
 PISTON59
 PISTON60
 PISTON61
 PISTON62
 PISTON63
 PISTON64
 PISTON65
 PISTON66
 PISTON67
 PISTON68
 PISTON69
 PISTON70
 PISTON71
 PISTON72
 PISTON73
 PISTON74
 PISTON75
 PISTON76
 PISTON77
 PISTON78
 PISTON79
 PISTON80
 PISTON81
 PISTON82
 PISTON83
 PISTON84
 PISTON85
 PISTON86
 PISTON87
 PISTON88
 PISTON89
 PISTON90
 PISTON91
 PISTON92
 PISTON93
 PISTON94
 PISTON95
 PISTON96
 PISTON97
 PISTON98
 PISTON99
 PISTO100
 PISTO101
 PISTO102
 PISTO103
 PISTO104
 PISTO105

PIS10106
 PIS10107
 PIS10108
 PIS10109
 PIS10110
 PIS10111
 PIS10112
 PIS10113
 PIS10114
 PIS10115
 PIS10116
 PIS10117
 PIS10118
 PIS10119
 PIS10120
 PIS10121
 PIS10122
 PIS10123
 PIS10124
 PIS10125
 PIS10126
 PIS10127
 PIS10128
 PIS10129
 PIS10130
 PIS10131
 PIS10132
 PIS10133
 PIS10134
 PIS10135
 PIS10136
 PIS10137
 PIS10138
 PIS10139
 PIS10140
 PIS10141
 PIS10142
 PIS10143
 PIS10144
 PIS10145
 PIS10146
 PIS10147
 PIS10148
 PIS10149
 PIS10150
 PIS10151
 PIS10152
 PIS10153
 PIS10154
 PIS10155
 PIS10156
 PIS10157

```

C      READ THE CODE  -1      ( END INDICATOR OF H-, DHDX- MATRICES)
C      READ      (NTPH )
C      READ      (NTPDH)
C      READ      (NTPH4)
C      IF      (JRUN .EQ. 1) WRITE (NTPH)  CODE
C
C      NTAPE      = NTEMP
C      NMTO       = NASYM
C      CONTINUE
C
C      190
C      PRESSURES DUE TO MODES COMPLETED  -- BEGIN CALCULATIONS DUE
C      TO GUSTS
C
C      DO 320 IR = 1, NR
C      COSA      = CR(1, IR)
C      COSB      = CR(2, IR)
C      COSC      = CR(3, IR)
C
C      R          = FMACH / (1.0 + FMACH*COSA)
C      TAU        = 2.0*KR * R / CBAK
C      LP         = 1
C      IS         = 1
C      NC         = NCARAY(LP)
C      NBXCUM     = NC
C
C      DO 300 I = 1, NBXC
C      DX18       = 0.5*(XC(2,I)+XC(4,I) - XC(1,I)-XC(3,I))
C      IGO        = 1
C      THETA      = -SG(IS)*COSB + CG(IS)*COSC
C      XLI        = XC(1, I)
C      XTI        = XC(2, I)
C      XLO        = XC(3, I)
C      XTO        = XC(4, I)
C      G          = CG(IS)*COSB + SG(IS)*COSC
C
C      IF (ABS(COSA) .LE. 0.001) GO TO 270
C      YMULT      = YS(IS) - EE(IS) * CG(IS)
C      ZMULT      = ZS(IS) - EE(IS) * SG(IS)
C      DLTRM      = 2.0*EE(IS) * (CG(IS)*COSB + SG(IS)*COSC)
C      ADDON      = YMULT*COSB + ZMULT*COSC
C
C      200
C      ELM1       = XLI * COSA + ADDON
C      ELM2       = XTI * COSA + ADDON
C      ARG31      = TAU * ((XLO - XLI) * COSA + DLTRM)
C      ARG42      = TAU * ((XTO - XTI) * COSA + DLTRM)
C
C      IF (ABS(ARG31) .LE. 0.01) GO TO 210
C      A31        = (COS(ARG31) - 1.0) / ARG31
C      B31        = SIN(ARG31) / ARG31
C      GO TO 220
C
C
C      PIST0158
C      PIST0159
C      PIST0160
C      PIST0161
C      PIST0162
C      PIST0163
C      PIST0164
C      PIST0165
C      PIST0166
C      PIST0167
C      PIST0168
C      PIST0169
C      PIST0170
C      PIST0171
C      PIST0172
C      PIST0173
C      PIST0174
C      PIST0175
C      PIST0176
C      PIST0177
C      PIST0178
C      PIST0179
C      PIST0180
C      PIST0181
C      PIST0182
C      PIST0183
C      PIST0184
C      PIST0185
C      PIST0186
C      PIST0187
C      PIST0188
C      PIST0189
C      PIST0190
C      PIST0191
C      PIST0192
C      PIST0193
C      PIST0194
C      PIST0195
C      PIST0196
C      PIST0197
C      PIST0198
C      PIST0199
C      PIST0200
C      PIST0201
C      PIST0202
C      PIST0203
C      PIST0204
C      PIST0205
C      PIST0206
C      PIST0207
C      PIST0208
C      PIST0209

```

```

210 CONTINUE = -ARG31 / 2.0
    A31 = 1.0
    B31 = 1.0
220 CONTINUE
    C IF (ABS(ARG42) .LE. 0.01) GO TO 230
    A42 = (COS(ARG42) - 1.0) / ARG42
    B42 = SIN(ARG42) / ARG42
    GO TO 240
230 CONTINUE = -ARG42 / 2.0
    A42 = 1.0
    B42 = 1.0
240 CONTINUE
    C COSIL1 = COS(TAU*ELM1)
    COSTL2 = COS(TAU*ELM2)
    SINTL1 = SIN(TAU*ELM1)
    SINTL2 = SIN(TAU*ELM2)
    C DCPTRM = CMPLX(( A31*COSIL1 - B31*SINTL1
    1 - A42*COSTL2 + B42*SINTL2)
    2 (-B31*COSIL1 - A31*SINTL1 + B42*COSTL2 + A42*SINTL2))
    C DENCM = DXIB * TAU * COSA
    GO TO (250,260), IGO
250 CONTINUE = THETA * FOUROM * DCPTRM / DENOM
    DCP1 = THETA * FOUROM * DCPTRM / DENOM
    C IGO
    THETA = 2*SG(IS)*COSB + CG(IS)*COSC
    YMULT = -YS(IS) - EE(IS)*CG(IS)
    ZMULT = ZS(IS) + EE(IS)*SG(IS)
    DLTRM = 2.0*EE(IS) * (CG(IS)*COSB - SG(IS)*COSC)
    XLI = XC(3, I)
    XTI = XC(4, I)
    XLO = XC(1, I)
    XTO = XC(2, I)
    GO TO 200
260 CONTINUE
    C DCP2 = THETA * FOUROM * DCPTRM / DENOM
    GO TO 280
270 CONTINUE
    LMBNOT = TAU * (YS(IS)*COSB + ZS(IS)*COSC)
    272 TEG = TAU * EE(IS) * G
    IF (ABS(TEG) .LE. 0.001) GO TO 278
    Q = SIN(TEG) / TEG
    TEGMLT = (COS(TEG) - Q) / TEG
274 CONTINUE

```

PISIO210
 PISIO211
 PISIO212
 PISIO213
 PISIO214
 PISIO215
 PISIO216
 PISIO217
 PISIO218
 PISIO219
 PISIO220
 PISIO221
 PISIO222
 PISIO223
 PISIO224
 PISIO225
 PISIO226
 PISIO227
 PISIO228
 PISIO229
 PISIO230
 PISIO231
 PISIO232
 PISIO233
 PISIO234
 PISIO235
 PISIO236
 PISIO237
 PISIO238
 PISIO239
 PISIO240
 PISIO241
 PISIO242
 PISIO243
 PISIO244
 PISIO245
 PISIO246
 PISIO247
 PISIO248
 PISIO249
 PISIO250
 PISIO251
 PISIO252
 PISIO253
 PISIO254
 PISIO255
 PISIO256
 PISIO257
 PISIO258
 PISIO259
 PISIO260
 PISIO261

```

C
1  EILNOT = CMPLX(COS(LMBNOT), -SIN(LMBNOT))
   DCPTRM = CMPLX(Q, ((XLI-XTI-XLO+XTU) / (XLI-XTI+XLO-XTU))
           * TEGMLT )
C
276 GO TO (276, 279), IGO
   CONTINUE = THETA * FOURUM * EILNOT * DCPTRM
   IGO = 2
   THETA = SG(IS)*COSB + CG(IS)*COSC
   G = CG(IS)*COSB - SG(IS)*COSC
   LMBNOT = TAU * (-YS(IS)*COSB + ZS(IS)*COSC)
   XLI = XC(3, I)
   XTI = XC(4, I)
   XLO = XC(1, I)
   XTU = XC(2, I)
   GO TO 272
C
278 Q = 1.0
   TEGMLT = 0.0
   GO TO 274
C
279 CONTINUE = THETA * FOURUM * EILNOT * DCPTRM
   DCP2
C
280 CONTINUE
   DCPS(I) = 0.5 * (DCP1 + DCP2)
   DCPA(I) = 0.5 * (DCP1 - DCP2)
C
   IF (I.LT. NBXCUM) GO TO 300
   IF (I.NE. NBARAY(LP)) GO TO 290
   LP = LP + 1
   NC = NBARAY(LP)
   IS = 1
   NBXCUM = NBXCUM + NC
C
290 IS
   NBXCUM = NBXCUM + NC
C
300 CONTINUE
   ONE SYMMETRIC, AND ONE ANTISYMMETRIC PRESSURE COLUMN DUE TO
   GUST COMPLETE, SAVE ON TAPE (AND PRINT)
C
   IF (IPRINT.LT. 2) GO TO 310
   WRITE (NTP6, 510) IR
   WRITE (NTP6, 480) (DCPS(I), I = 1, NBOX)
   WRITE (NTP6, 520) IR
   WRITE (NTP6, 480) (DCPA(I), I = 1, NBOX)
C
310 CONTINUE
   WRITE (MASTCP) (DCPS(I), I = 1, NBOX)
   WRITE (NTEMP) (DCPA(I), I = 1, NBOX)
C
320 CONTINUE
   PRESSURES DUE TO GUSTS COMPLETE --- COMPUTE BODY FORCES
   DUE TO GUSTS
   THE VARIABLES DCP1, DCP2, DCPS, DCPA ARE USED HERE FOR THE
   TEMPORARY STORAGE OF THE V-FORCE COMPONENTS

```

PIST0262
 PIST0263
 PIST0264
 PIST0265
 PIST0266
 PIST0267
 PIST0268
 PIST0269
 PIST0270
 PIST0271
 PIST0272
 PIST0273
 PIST0274
 PIST0275
 PIST0276
 PIST0277
 PIST0278
 PIST0279
 PIST0280
 PIST0281
 PIST0282
 PIST0283
 PIST0284
 PIST0285
 PIST0286
 PIST0287
 PIST0288
 PIST0289
 PIST0290
 PIST0291
 PIST0292
 PIST0293
 PIST0294
 PIST0295
 PIST0296
 PIST0297
 PIST0298
 PIST0299
 PIST0300
 PIST0301
 PIST0302
 PIST0303
 PIST0304
 PIST0305
 PIST0306
 PIST0307
 PIST0308
 PIST0309
 PIST0310
 PIST0311
 PIST0312
 PIST0313

PIST0314
 PIST0315
 PIST0316
 PIST0317
 PIST0318
 PIST0319
 PIST0320
 PIST0321
 PIST0322
 PIST0323
 PIST0324
 PIST0325
 PIST0326
 PIST0327
 PIST0328
 PIST0329
 PIST0330
 PIST0331
 PIST0332
 PIST0333
 PIST0334
 PIST0335
 PIST0336
 PIST0337
 PIST0338
 PIST0339
 PIST0340
 PIST0341
 PIST0342
 PIST0343
 PIST0344
 PIST0345
 PIST0346
 PIST0347
 PIST0348
 PIST0349
 PIST0350
 PIST0351
 PIST0352
 PIST0353
 PIST0354
 PIST0355
 PIST0356
 PIST0357
 PIST0358
 PIST0359
 PIST0360
 PIST0361
 PIST0362
 PIST0363
 PIST0364
 PIST0365

```

C      DO 430 IR = 1, NR
      COSA = CR(1, IR)
      COSB = CR(2, IR)
      COSC = CR(3, IR)
      R = FMACH / (1.0 + FMACH*COSA)
      LB2 = 0

C      DO 410 LB = 1, NB
      THETAZ = COSC

C      LB1 = LB2 + 1
      LB2 = LB2 + NSBEA(LB)
      DO 400 I = 1, LB1, LB2
      IZY = 1
      THE TAY = COSB
      IF (YB(LB) .EQ. 0.0) GO TO 330
      THEIAY = YB(LB)*COSB / ABS(YB(LB))
      330 CONTINUE

C      YTRM = YB(LB) * COSB
      ZTRM = ZB(LB) * COSC
      ADDON = YTRM + ZTRM
      DELX = XIS2(I) - XIS1(I)
      X0 = 0.5 * (XIS1(I) + XIS2(I))
      BJ = DELX * R * COSA / CBAR
      KBJ = KR * BJ
      C1 = 2.0 * PI * AO(I) * (1.0 - R * COSA) / FMACH
      C2 = -PI * CBAR * (AO(I) - 0.5 / FMACH) / FMACH
      IF (KR .NE. 0.0) C2OKR = C2 / KR
      TERM1 = 1.0
      IF (ABS(KBJ) .LE. 0.001) GO TO 340
      TERM1 = SIN(KBJ) / KBJ
      340 CONTINUE

C      350 CONTINUE = 2.0 * R * (X0 * COSA + ADDON) / CBAR
      AJ = AJ * KR
      ELAK = CMPLX( COS(AJK), -SIN(AJK) )
      TERM2 = CMPLX ( C1, C2OKR )
      FPART = ELAK * TERM1 * TERM2
      GU TO (360, 380), IZY
      DFQ1 = THE TAY * FPART
      DCPI = THEIAY * FPART

C      IF (YB(LB) .EQ. 0.0) GO TO 370
      YTRM = YB(LB) * COSB
      ADDON = YTRM + ZTRM
      THE TAY = YTRM / ABS(YB(LB))
      IZY = 2
      GO TO 350
  
```



```

REWIND NTEMP
IF (NB.EQ.0) GO TO 590
IF (NASYM.EQ.0) GO TO 590
NTAPE = MASTFZ
DO 580 IZY = 1, 2
DO 570 NM = 1, NASYM
  REAC (NTEMP) (DFQA(I), I = 1, NSBETO)
  WRITE (NTAPE) (DFQA(I), I = 1, NSBETO)
570 CONTINUE
C
  NTAPE = MASTFY
580 CONTINUE
590 CONTINUE
C
  NMTD = NASYM + NR
DO 600 NM = 1, NMTD
  REAC (NTEMP) (DCPA(I), I = 1, NBOX)
  WRITE (MASTCP) (DCPA(I), I = 1, NBOX)
600 CONTINUE
C
  IF (NB.EQ.0) GO TO 630
DO 610 NM = 1, NR
  REAC (NTEMP) (DFQA(I), I = 1, NSBETO)
  WRITE (MASTFZ) (DFQA(I), I = 1, NSBETO)
  REAC (NTEMP) (DFQA(I), I = 1, NSBETO)
  WRITE (MASTFY) (DFQA(I), I = 1, NSBETO)
610 CONTINUE
C
630 CONTINUE
C
  RETURN
C
  END

```

PIST0418
 PIST0419
 PIST0420
 PIST0421
 PIST0422
 PIST0423
 PIST0424
 PIST0425
 PIST0426
 PIST0427
 PIST0428
 PIST0429
 PIST0430
 PIST0431
 PIST0432
 PIST0433
 PIST0434
 PIST0435
 PIST0436
 PIST0437
 PIST0438
 PIST0439
 PIST0440
 PIST0441
 PIST0442
 PIST0443
 PIST0444
 PIST0445
 PIST0446
 PIST0447
 PIST0448
 PIST0449
 PIST0450

```

SUBROUTINE READD ( D, N, NTAPE )
  SUBROUTINE TO READ DZ OR DY FROM TAPE
  USE OF AN ARRAY TYPE OF INPUT TO INCREASE SPEED
  COMPLEX D(N)
  READ ( NTAPE ) IROW, IT, D
  RETURN
END

```

```

READD
READD
READD
READD
READD
READD

```

2 3 4 5 6 7 8

RHSIDE 2
 RHSIDE 3
 RHSIDE 4
 RHSIDE 5
 RHSIDE 6
 RHSIDE 7
 RHSIDE 8
 RHSIDE 9
 RHSIDE 10
 RHSIDE 11
 RHSIDE 12
 RHSIDE 13
 RHSIDE 14
 RHSIDE 15

```

C      SUBROUTINE RHSIDE( NTOT, NMODE, JM, IX, MK1, MK2, LENGTH,
1      COMPLEX WORK(LENGTH), RHS(NMODE)
      DO 10 JX=1,NMODE
      JM = JM+1
      JJ = (JM - 1) * NTOT + IX
      RHS(JX) = WORK(JJ)
      IF (IX.GE.MK1. AND .IX.LE.MK2) RHS(JX) = (0.0,0.0)
      10 CONTINUE
C      RETURN
      END

```

```

SUBROUTINE ROWDYZ (NFB, DA, NLB, DX, ROW, DZ, IRB,
1  BETA, IDZDY, NTAPE, NTAPE2, SGR, CGR, IPRNT
2  NSBE, AO, YB, ZB, AR, XIS1, XIS2)
3  DIMENSION XIS1(1), XIS2(1)
4  DIMENSION XIS1(1), XIS2(1)
5  CALCULATE A ROW OF DZ OR DY
6  SLENDER BODY
7  CROWDYZ
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

ROWDY250
 ROWDY251
 ROWDY252
 ROWDY253
 ROWDY254
 ROWDY255
 ROWDY256
 ROWDY257
 ROWDY258
 ROWDY259
 ROWDY260
 ROWDY261
 ROWDY262
 ROWDY263
 ROWDY264
 ROWDY265
 ROWDY266
 ROWDY267
 ROWDY268
 ROWDY269
 ROWDY270
 ROWDY271
 ROWDY272
 ROWDY273
 ROWDY274
 ROWDY275
 ROWDY276
 ROWDY277
 ROWDY278
 ROWDY279
 ROWDY280
 ROWDY281
 ROWDY282
 ROWDY283
 ROWDY284
 ROWDY285
 ROWDY286
 ROWDY287
 ROWDY288
 ROWDY289
 ROWDY290
 ROWDY291
 ROWDY292
 ROWDY293
 ROWDY294
 ROWDY295
 ROWDY296
 ROWDY297
 ROWDY298
 ROWDY299
 ROWDY300
 ROWDY301

```

C      D( 1, T1)      =      0.0
C      D( 2, T1)      =      0.0
C      DA(1, T1)      =      0.0
C      DA(2, T1)      =      0.0
C      LHS              =      0
C      X11              =      XIS1( T1 )
C      X12              =      XIS2( T1 )
C      AZRO              =      AO( T1 )
C      ETA              =      YB( B )
C      ZETA              =      ZB( B )
C      CHECK TO SEE IF CALCULATIONS ARE TO BE MADE
C      IF ( ( DY .EQ. ETA ) .AND. ( DZ .EQ. ZETA ) ) GO TO 20
C
C      ASSIGN 10 TO JDZDY
C      GO TO 110
C      D( 1, T1)      =      DZ
C      D( 2, T1)      =      DZY
C      DA(1, T1)      =      DZ
C      DA(2, T1)      =      DZY
C
C      20 CONTINUE
C      IF ( DELTA .EQ. 0.0 ) GO TO 60
C      ETA              =      -YB( B )
C      IF SENDING BODY LIES ON Z AXIS.. SKIP SYMMETRY
C      IF ( ETA .EQ. 0.0 ) GO TO 60
C      CHECK TO SEE IF CALCULATIONS ARE TO BE MADE
C      IF ( ( DY .EQ. ETA ) .AND. ( DZ .EQ. ZETA ) ) GO TO 40
C      LHS              =      1
C      RETURN ADDRESS FROM DZY
C
C      ASSIGN 30 TO JDZDY
C      GO TO 110
C      D( 1, T1)      =      D(1, T1) + DELTA * DZ
C      DA(1, T1)      =      DA(1, T1) + DELTA * DZY
C      D( 2, T1)      =      D(2, T1) + DELTA * DZ
C      DA(2, T1)      =      DA(2, T1) + DELTA * DZY
C
C      40 CONTINUE
C      IF ( EPSILON .EQ. 0.0 ) GO TO 80
C      CALC. ONLY IF DELTA AND EPSILON NOT EQUAL ZERO
C      ETA              =      -YB( B )
C      ZETA              =      -ZB( B )
C      IF SENDING BODY LIES ON Z OR Y AXIS.. SKIP GRND. AND SYMMETRY
C      IF ( ZETA .EQ. 0.0 ) GO TO 60
C      CHECK TO SEE IF CALCULATIONS ARE TO BE MADE
C      IF ( ( DY .EQ. ETA ) .AND. ( DZ .EQ. ZETA ) ) GO TO 60
C      LHS              =      1
C      ASSIGN RETURN ADDRESS FROM DZY
C
C      ASSIGN 50 TO JDZDY
C      GO TO 110
C      D( 1, T1)      =      D( 1, T1 ) + EPSILON * DELTA * DZ
  
```

```

C      DA(1, T1) = DA(1, T1) + EPSLON* DELT2 * DZVR
C      D(2, T1) = D(2, T1) + EPSLON* DELTA * DZVI
C      DA(2, T1) = DA(2, T1) + EPSLON* DELT2 * DZVI

C      60 IF ( EPSLON .EQ. 0.0 ) GO TO 80
C      ETA = YB(B)
C      ZETA = -ZB(B)
C      IF SENDING BODY LIES ON THE Y AXIS.. SKIP GROUND EFFECTS
C      IF ( ZETA .EQ. 0.0 ) GO TO 80
C      CHECK TO SEE IF CALCULATIONS ARE TO BE MADE
C      IF ( ( DY .EQ. ETA ) .AND. ( DZ .EQ. ZETA ) ) GO TO 80
C      ASSIGN RETURN ADDRESS FROM DZY

C      ASSIGN 70 TO JZDY
C      GO TO 110
C      70 D(1, T1) = D(1, T1) + EPSLON * DZVR
C      DA(1, T1) = DA(1, T1) + EPSLON * DZVR
C      D(2, T1) = D(2, T1) + EPSLON * DZVI
C      DA(2, T1) = DA(2, T1) + EPSLON * DZVI
C      80 CONTINUE
C      END OF LOOP FOR ELEMENT
C      90 CONTINUE
C      200 IS END OF LOOP ON SLENDER BODY
C      100 CONTINUE
C      WRITE (NTAPE) ROW, T1, D
C      WRITE (INTAP2) ROW, T1, DA
C      IF ( IPRNT .NE. 0 ) WRITE (NPOT,120) ROW, T1, D
C      RETURN

C      CALL SEQUENCE TO DZY
C      110 CALL DZY ( DX, CY, X12, REFC, DZ, SGR, ZETA, CGR,
C      1 AZRC, KR, IDZDY, DZVR, LHS,
C      2 FMACH,
C      3 IPRNT = 0
C      4 LHS = 0

C      GO TO JZDY, (10,30,50,70)
C      120 FORMAT (18H ROWDYZ--- RCW NO., I5, 1H,, 110,10H ELEMENTS. /
C      1 ( 6E12.4) )
C      130 FORMAT (12H ROWDYZ B =, I10, 4E20.8 )
C      140 FORMAT (1H, Z10 )
C      END

```

```

ROWDY102
ROWDY103
ROWDY104
ROWDY105
ROWDY106
ROWDY107
ROWDY108
ROWDY109
ROWDY110
ROWDY111
ROWDY112
ROWDY113
ROWDY114
ROWDY115
ROWDY116
ROWDY117
ROWDY118
ROWDY119
ROWDY120
ROWDY121
ROWDY122
ROWDY123
ROWDY124
ROWDY125
ROWDY126
ROWDY127
ROWDY128
ROWDY129
ROWDY130
ROWDY131
ROWDY132
ROWDY133
ROWDY134
ROWDY135
ROWDY136
ROWDY137
ROWDY138
ROWDY139
ROWDY140
ROWDY141
ROWDY142
ROWDY143
ROWDY144
ROWDY145
ROWDY146
ROWDY147
ROWDY148
ROWDY149
ROWDY150
ROWDY151

```



```

C C
REWIND NW5
REWIND NW6
C C
CALCULATE DZ AND DY MATRICES
C C
REWIND NDZ
REWIND NDZ2
C
IDZY = 0
LOC = NTYS + 1
CALL DZYMAT(WORK(1), WORK(LOC), NFZB, NLZB, NTYS, IDZY,
1 X , BETA)
2 , NUMBR NSBE,NC,NS,AO,YB,ZB,AR,XIS1,XIS2,CG,SG,YP,ZP )
2 , NREA,
C C C
CALCULATE DY
C C
REWIND NDY
REWIND NDY2
IDZY = 1
LOC = NTYS + 1
CALL DZYMAT(WORK(1), WORK(LOC), NFYB, NLVB, NTYS, IDZY,
1 X , BETA)
2 , NUMBR NSBE,NC,NS,AO,YB,ZB,AR,XIS1,XIS2,CG,SG,YP,ZP )
2 , NBEA,
C C C
10 CONTINUE
NTZY = NTYS + NTYS
NFW = 1
NFDZ = NFW + N * NTYS
NFUY = NFDZ + N * NTYS
NFUZ = NFUY + N * NTYS
NFCPZ = NFUY + NTYS
NFCPY = NFCPZ + NTYS
NFF = NFCPY + NTYS
IEND = NFCPY - 1
NFE = NFE + NTYS
NFDH = NFW + N
C
IGD = 1
ASSIGN 70 TO IZ
ASSIGN 130 TO IY
20 CONTINUE = 0
J
REWIND NDZ
REWIND NDY
C C
READ DZ AND DY FROM TAPE
N1 = NFDZ
DO 30 I = 1, N

```

```

102 CALL READD ( WORK(N1), NTZS , NDZ )
103 N1 = N1 + NTZS
104 N1 CONTINUE = NFDY
105 N1 = 40
106 I = 1, N
107 CALL READD ( WORK(N1), NTYS , NDY )
108 N1 = N1 + NTYS
109 N1 CONTINUE =
110 N1 CONTINUE
111
112 NOW CALCULATE UZ AND UY A COLUMN AT A TIME AND
113 MULTIPLY
114
115 DO 50 IWK = NFUZ , IEND
116 WORK(IWK) = 0.0
117 CONTINUE
118
119 NFEND = MAXO(NTZS, NTYS)
120 CALL ZEROUT(WORK(NFE), NFEND, 1, 0 )
121 CONTINUE
122 READ (NTPH) NODE
123 IF (NODE.EQ.-1) GO TO 150
124 J = J + 1
125 GO TO 12 , (70,90)
126
127 CALCULATE UZ COLUMN
128
129 70 CONTINUE
130 CALL MUZYC ( J , NSBE , 2 , NTZS , NFZB , NLZB ,
131 1 NSBE , KR , XIS1 , XIS2 , AO ,
132 2 NTPDH, WORK(NFUZ), WORK(NFCPZ),
133 3 UZYA, CPZYA, NR, FMACH, YB, ZB, CR, NG, WORK(NFH), WORK(NFDH) ,
134 IEND = NFE - 1
135
136 CALL ZEROUT(WORK(NFZB), N , 1, 0 )
137 DUMULT(1, N, NTZS, 0 , WORK(1),
138 1 WORK(NFDZ), WORK(NFUZ), 0 , 0 )
139
140 WRITE (NW3 ) NODE
141 CALL RWREC( 0, NW3, WORK , N , 1, 0 )
142 NFEND = NFCPZ + NTZS - 1
143 JJEND = NFE + NTYS - 1
144 NFUZYL = NFUY - 1
145 WRITE (NW5) ((WORK(I1), I1 = NFCPZ, NFEND),
146 1 ((WORK(JJ), JJ = NFE, JJEND))
147
148 CALCULATIONS FOR UY COLUMN
149
150 90 GO TO IY, (100,130)
151
152 100 CONTINUE
153

```

```

110 CALL MUZYC ( J, NSBE, 3, NTYS, NLYB,
111     NTPDH, WORK(NFYU), KR, XIS1, NLYB,
112     UZYA, CPZYA, NR, FMACH, YB, ZB, CR, NG, WORK(NFH), WORK(NFDH) )
113     CALL
114     CALL
115     CALL
116     CALL
117     CALL
118     CALL
119     CALL
120     CALL
121     CALL
122     CALL
123     CALL
124     CALL
125     CALL
126     CALL
127     CALL
128     CALL
129     CALL
130     CALL
131     CALL
132     CALL
133     CALL
134     CALL
135     CALL
136     CALL
137     CALL
138     CALL
139     CALL
140     CALL
141     CALL
142     CALL
143     CALL
144     CALL
145     CALL
146     CALL
147     CALL
148     CALL
149     CALL
150     CALL
151     CALL
152     CALL
153     CALL
154     CALL
155     CALL
156     CALL
157     CALL
158     CALL
159     CALL
160     CALL
161     CALL
162     CALL
163     CALL
164     CALL
165     CALL
166     CALL
167     CALL
168     CALL
169     CALL
170     CALL
171     CALL
172     CALL
173     CALL
174     CALL
175     CALL
176     CALL
177     CALL
178     CALL
179     CALL
180     CALL
181     CALL
182     CALL
183     CALL
184     CALL
185     CALL
186     CALL
187     CALL
188     CALL
189     CALL
190     CALL
191     CALL
192     CALL
193     CALL
194     CALL
195     CALL
196     CALL
197     CALL
198     CALL
199     CALL
200     CALL
201     CALL
202     CALL
203     CALL
204     CALL
205     CALL

```

```

206 REWIND NW5
207 REWIND NW6
208 DO 300 MD = 1, NM
209 READ (NW3) JCOL
210 READ (NW3) ( DW (L), L = 1, N)
211 RFAC (NW4)
212 READ (NW4) (WORK(L), L = 1, N)
213 DO 280 L = 1, N
214 DW(L) = OW(L) + WORK(L)
215 CONTINUE
216 C
217 READ (NW5) (CPZYA(L), L=1, NTZY)
218 READ (NW6) (WORK(L), L = 1, NTZY)
219 DO 290 L = 1, NTZY
220 CPZYA(L) = CPZYA(L) + WORK(L)
221 CONTINUE
222 WRITE (NW1) JCOL
223 WRITE (NW1) ( DW (L), L = 1, N)
224 WRITE (NW2) (CPZYA(L), L=1, NTZY)
225 IF (MD .NE. NSYM) GO TO 270
226 READ (NW3) CODE
227 READ (NW4)
228 WRITE (NW1) CODE
229 CONTINUE
230 C
231 IF (IPRINT .LT. 2) GO TO 300
232 WRITE (NTP6,310) MD, NTZY, (CPZYA(L), L=1,NTZY)
233 C
234 CONTINUE
235 IF (NASYM .NE. 0) WRITE (NW1) CODE
236 C
237 REWIND NW3
238 REWIND NW5
239 REWIND NW6
240 NDY = NDZOLD
241 NDY = NCYOLD
242 IGO = 1
243 L2 = NTZS + 1
244 C
245 CONTINUE
246 REWIND NDZ
247 REWIND NDY
248 NI = NFDZ
249 DO 460 I = 1, N
250 CALL READD(WORK(NI), NTZS, NDZ)
251 NI = NI + NTZS
252 CONTINUE
253 NI = NFDY
254 DO 480 I = 1, N
255 CALL READD(WORK(NI), NTYS, NDY)
256 NI = NI + NTYS
257 CONTINUE
258 C
259 480 CONTINUE

```

```

C      DO 560 NR = 1, NG
      J = J + 1
      CALL ZEROUT(WORK(NFZB), N, 1, 0)
      GO TO (500, 520), IGO
500  CONTINUE
      CALL MUZYC( J, 2, NTZS, NFZB, NSBE, KR, REFC,
1      AO, AOP, XIS1, XIS2, AR, NTPH, NTPDH
2      WORK(NFUZ), WORK(NFCPZ), UZYA( 1 ), CPZYA( 1 )
3      NR, FMACH, YB, ZB, CK, NG, WORK(NFH), WORK(NFDH) )
C
      CALL MUZYC( J, 3, NTYS, NFYB, NSBE, KR, REFC,
1      AO, AOP, XIS1, XIS2, AR, NTPH, NTPDH
2      WORK(NFUY), WORK(NFCPY), UZYA( L2 ), CPZYA( L2 )
3      NR, FMACH, YR, ZB, CK, NG, WORK(NFH), WORK(NFDH) )
C
      CALL ZEROUT(WORK(NFZR), N, 1, 0)
C
      CALL DUMULT( 1, N, NTZS, NTYS, WORK( 1 ),
1      WORK(NFDZ), WORK(NFUZ), WORK(NFDY), WORK(NFUY) )
C
      WRITE (NW1) (WORK(LW), LW = 1, N)
      WRITE (NW3) (WORK(LC), LC = NFCPZ, NFCEND)
      WRITE (NW5) (UZYA( L ), L = 1, NTZY)
      WRITE (NW6) (CPZYA(L), L = 1, NTZY)
      GO TO 560
520  CONTINUE
C
      READ (NW5) (UZYA( L ), L = 1, NTZY)
C
      CALL DUMULT( 1, N, NTZS, NTYS, WORK( 1 ),
1      WORK(NFDZ), UZYA( 1 ), WORK(NFDY), UZYA( L2 ) )
C
      WRITE (NW1) (WORK(LW), LW = 1, N)
560  CONTINUE
      IF (IGO.EQ. 2 .OR. NASYM.EQ. 0) GO TO 700
      REWIND NW5
      NDZ = NDZ2
      NDY = NDY2
      IGO = 2
      GO TO 450
C
700  CONTINUE
C
      REWIND NW2
      REWIND NW3
      REWIND NW5
      REWIND NW6
      CALL RWREC( 2 , NW2, CPZYA, NTZY, NM , NW5 )
      REWIND NW2
      REWIND NW5
      CALL RWREC( 2 , NW5, CPZYA, NTZY, NSYM , NW2 )

```

SB 258
 SB 259
 SB 260
 SB 261
 SB 262
 SB 263
 SB 264
 SB 265
 SB 266
 SB 267
 SB 268
 SB 269
 SB 270
 SB 271
 SB 272
 SB 273
 SB 274
 SB 275
 SB 276
 SB 277
 SB 278
 SB 279
 SB 280
 SB 281
 SB 282
 SB 283
 SB 284
 SB 285
 SB 286
 SB 287
 SB 288
 SB 289
 SB 290
 SB 291
 SB 292
 SB 293
 SB 294
 SB 295
 SB 296
 SB 297
 SB 298
 SB 299
 SB 300
 SB 301
 SB 302
 SB 303
 SB 304
 SB 305
 SB 306
 SB 307
 SB 308
 SB 309

310
311
312
313
314
315
316
317
318
319
320
321

SH
SB
SB
SB
SB
SB
SB
SB
SB
SB

```

CALL RWREC(2,NW3,CPZYA,NTZY,NG,NW2)
IF (NASYM.EQ.0) GO TO 710
CALL RWREC(2,NW5,CPZYA,NTZY,NASYM,NW2)
CALL RWREC(2,NW6,CPZYA,NTZY,NG,NW2)
CONTINUE
710 CONTINUE
REWIND NW1
REWIND NW2
310 FORMAT ( // 9H MODE =, 15 / 16H NTZS + NTYS =, 15 /
CP-Z*DELTA-A, CP-Y*DELTA-A // 16E14.6 ) )
C
RETURN
END

```

```

SUBROUTINE SOLM (ELXI,ELYI,ELZI,PHIN,PHIZ,PHIY, NODES
1 , NSYM, NASB, NFIX1, MFIY2, IFNEWH, MDOF, PHI, MDOFP, MDOFB
2 , NAS, NASB, NBARAY, NCARAY, NSARAY, ISSR, NSSTR
3 , IFLA, NBEA, NT12, NSBEA
4 , YB, ZB, ARB, AVR, XLE, XTE, RIA, THIA, AO, AOP, XIS1, XIS2
5 , CG, CS, EE, SG, YS, ZS, XIJ, YIN, ZIN, COORD
6 , X, XIC, DELX, XLAM, H, DHDX, DELA
7 , DT, DTA, RHS, XC )
**
SUBSONIC UNSTEADY AERODYNAMICS FOR GENERAL CONFIGURATIONS
METHOD CALCULATES LIFTING PRESSURE, SPAN LOADS AND MOMENTS,
GENERALIZED FORCES, AND OTHER AERODYNAMIC CHARACTERISTICS FOR
GENERAL LIFTING SURFACE - BODY COMBINATIONS. STEADY AND UNSTEADY
FLOWS ARE CONSIDERED. BODIES MAY HAVE ELLIPTIC CROSS SECTIONS.
LIFTING SURFACES MAY HAVE ARBITRARY PLANEFORM AND DIHEDRAL.
CONTROL SURFACES ARE HANDLED. PROGRAM IS LARGE ENOUGH TO HANDLE
ENTIRE AIRCRAFT.
**
COMMON NAA, WORK(1)
COMMON /AROCOM/ NTI, MODES
X , NP, MSTRIP, NSMAX, NCMAX, NTOTAL, NB, MSBE, MBE
Y , ND, NE, NBY, NBZ, NTO, NTP, NTY, NTZ
1 , NTYS, NTZS, MAXGR, MAXSTR, NSBETO, NSTRIP, KR, XM, REFA, REFC
2 , REFS, FMACH, LINES
REAL
KR
DIMENSION ELXI(1), ELYI(1), ELZI(1)
DIMENSION PHIN(NODES,1), PHIZ(NODES,1), PHIY(NODES,1)
DIMENSION PHI(1)
DIMENSION NAS(1), NASB(1), NBARAY(1), NCARAY(1), NSARAY(1)
DIMENSION ISSR(1), NSSTR(1)
DIMENSION IFLA(2,1), NBEA(2,1), AVR(1), XLE(1), XTE(1)
DIMENSION YB(1), ZB(1), THIA(1), TH2A(1)
DIMENSION RIA(1), AOP(1), XIS1(1), XIS2(1)
DIMENSION AO(1), CS(1), EE(1), SG(1), YS(1), ZS(1)
DIMENSION XIJ(1), YIN(1), ZIN(1), COORD(1)
DIMENSION X(1), XIC(1), DELX(1), XLAM(1)
DIMENSION H(1), DHDX(1), DELA(1)
COMPLEX DT(1), DIA(1), RHS(1)
DIMENSION FREQ(100)
DIMENSION IHD(50), RHD(50)
EQUIVALENCE (IHD(1), RHD(1))
DIMENSION NMODES(6), CR(3,20), NTAPE(20)
DIMENSION XC(1)
DATA NTAPE/1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 6*0 /
DATA NS8MIN, IWRITE, IZERO, 1, 0, 0 /
DATA NSBMIN, NUMBR, IZERO, 1, 0, 0 /
DATA NTPH, NTPDH, 12, 13, 14, 15, 16, 17, 18, 19, 20 /
DATA NTAERO, NTGF, 12, 13, 14, 15, 16, 17, 18, 19, 20 /
DATA MASTH, MASTDH, MASTH4, 21, 22, 23 /
DATA NDMINB, NDMINP, NModes, 1, 1, 6*0 /
1
2
DATA
DATA

```


102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153

```

NM      = NTAPE(1)
ND      = NTAPE(2)
NDW     = NTAPE(3)
NWT     = NTAPE(4)
NWT A   = 15
NI      = NTAPE(10)
NBFM    = NTAPE(11)
NBFMA   = NTAPE(12)
NEWBFZ  = NTAPE(13)
NEWBFY  = NTAPE(14)
NW      = NPTAP
NE      = 0
C      IF (IFNEWH.NE.0) GO TO 300
C
CALL HEADNG
WRITE (NOUT,50)
50  FORMAT (1H0,4HNODE,8X7HELXI(1),8X7HELXI(1),8X7HELXI(1))
DO 170 I=1,NODES
170 READ (NTI,20) ELXI(1), ELXI(1), ELXI(1)
WRITE (NOUT,40) I, ELXI(1), ELXI(1), ELXI(1)
40  FORMAT (1X,14,3E15.6)
C
DO 185 J=1,MODES
C
CALL HEADNG
WRITE (NOUT,60) J,J,J
60  FORMAT (1H0,4HNODE,5X7HPHIN(I,12,1H),5X7HPHIZ(I,12,1H)
1, 5X7HPHY(I,12,1H))
DO 180 I=1,NODES
180 READ (NTI,20) PHIN(I,J), PHIZ(I,J), PHIZ(I,J), PHIZ(I,J)
185 CONTINUE
C
200 CONTINUE
C
READ (NTI,10) IPR1, IPR2, IPR3, NGUST
READ (NTI,10) NKD, NKP, MK1, MK2
NK      = NKD+NKP
JRUN   = 0
IOPT   = 0
READ (NTI,20) FMACH, REFA, REFS, REFC, XM, SCALER
IF (SCALER.EQ.0.0) SCALER=2.0
READ (NTI,20) (FREQ(I),I=1,NK)
C
CALL HEADNG
WRITE (NOUT,90)
WRITE (NOUT,150) (FREQ(I), I = 1, NK)
BETA2  = 1.0 - FMACH**2
IF (BETA2.LE.0.0) BETA2 = ABS(BETA2)
BETA   = SQRT( BETA2 )
C

```

154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205

WRITE (NOUT,110) REFC, NP, NB, REFS, REFA, FMACH, BETA,

1 XM, SCALER, NP, NB, REFS, REFA, FMACH, BETA,
NG = I3
IF (NGUST.NE.0) NG = NGUST
IF (MFIX2.GT.NSYM) MFIX2 = MFIX2+NG
IF (MFIX1.GT.NSYM) MFIX1 = MFIX1+NG
WRITE(NOUT,91) IPR1, IPR2, IPR3, NG
FORMAT(1H0,20X, IPR1 = ,12/1H,
1 20X, IPR2 = ,12/1H,
1 20X, IPR3 = ,12/1H,
1 20X, NO. GUST ORIENTATIONS = ,I3)

IHD(2) = NK
IHD(3) = NSYM
IHD(4) = NASYM
IHD(5) = NG
IHD(6) = NB
IHD(11) = NP
RHD(21) = FMACH
RHD(22) = REFA
RHD(23) = REFS
RHD(24) = REFC
RHD(25) = XM

L1 = NAA
L2 = L1+NCMAX
L3 = L2+NSMAX
L4 = L3+MBE
L5 = L4+MBE
L6 = L5+MSBE
L7 = L6+MSBE
L8 = L7+NP
L9 = L8+MS
L10 = L9+MSTRIP+NB
L11 = L10+MSTRIP+NB
L12 = L11+NTOTAL
L13 = L12+NTOTAL
L14 = L13+NTOTAL
L15 = L14+NTOTAL
L16 = L15+NTOTAL
L17 = L16+NTOTAL
L18 = L17+NTOTAL
L19 = L18+NTOTAL
L20 = L19+MSBE
L21 = L20+MSBE

CALL GEOM (DELA, IFLA, NBEA, NT12, NAS, NASB, ISSTR, NSSTR
1, NSBEA, NBARAY, NCARAY, NSARAY, AO, YB, ZB, ARB, AVR, AOP, RIA
2, XLE, XTE, TH1A, XIS1, XIS2, X, CG, CS, EE, SG, YS, ZS, XIC, XIJ
3, YIN, ZIN, DELX, XLAM, COORD, NOUT
4, WORK(L1), WORK(L2), WORK(L3), WORK(L4), WORK(L5), WORK(L6)

SDLM 206
SDLM 207
SDLM 208
SDLM 209
SDLM 210
SDLM 211
SDLM 212
SDLM 213
SDLM 214
SDLM 215
SDLM 216
SDLM 217
SDLM 218
SDLM 219
SDLM 220
SDLM 221
SDLM 222
SDLM 223
SDLM 224
SDLM 225
SDLM 226
SDLM 227
SDLM 228
SDLM 229
SDLM 230
SDLM 231
SDLM 232
SDLM 233
SDLM 234
SDLM 235
SDLM 236
SDLM 237
SDLM 238
SDLM 239
SDLM 240
SDLM 241
SDLM 242
SDLM 243
SDLM 244
SDLM 245
SDLM 246
SDLM 247
SDLM 248
SDLM 249
SDLM 250
SDLM 251
SDLM 252
SDLM 253
SDLM 254
SDLM 255
SDLM 256
SDLM 257

```

5  WORK(L7),WORK(L8),WORK(L9),WORK(L10)
6  WORK(L11),WORK(L12),WORK(L13),WORK(L14),WORK(L15),WORK(L16)
7  XC
8  WORK(L17),WORK(L18),WORK(L19),WORK(L20)
9  IHD,WORK(L21),WORK(L21)

```

C

NTOT IS TOTAL NO. OF BOXES + 2* NO. OF INTERFERENCE ELEMENTS

```

NTOT = NTP + NTZ + NTY
MTOT = NTOT
NDTOB = 0
NDTOP = 0
N1 = NAA
N2 = N1 + NB
N3 = N2 + MAXO(NB,NP)
N4 = N3 + NODES
N5 = N4 + NODES
N6 = N5 + NODES
N8 = N6 + MSTRIP + NB
N9 = N8 + NTOTAL
N10 = N9 + NTOTAL
N11 = N10 + NTOTAL
L1 = N11 + NTOTAL

```

C

```

CALL SPLINE (NTPH,NTPDH,NTPH4,NDTOB,NDTOP,
1  ELXI, ELZI, ELZI, WORK(L1))
2  NSBEA, NBARAY, NCARAY, NSARAY, XIS1, XIS2, X, CG, SG, YS, ZS
3  XIC, XIJ, PHI, PHIZ, PHIY, MDOF, NODES, MDOF8, IFNEWH
4  WORK(N1), WORK(N2), WORK(N3), WORK(N4), WORK(N5), WORK(N6)
5  WORK(N8), WORK(N9), WORK(N10), WORK(N11), H, DHDX )
NMODES(1) = NSYM
IF (NB.NE. 0) NMODES(2) = NSYM
NMODES(3) = NMODES(2)
NMODES(4) = NASYM
IF (NB.NE. 0) NMODES(5) = NASYM
NMODES(6) = NMODES(5)
NSLEND = NSBETO + NB
NTMAX = MAXO(NTOT, NSLEND)
L1 = NAA
L2 = L1 + NTMAX
REWIND MASTH
REWIND MASTDH
REWIND MASTH4
CALL COTP (NTPH,NTPDH,NTPH4,MASTH,MASTDH,MASTH4, IFNEWH,
1  NSYM,NASYM,MDOF8,MODES,NB,NTMAX,
2  PHIN,PHIZ,PHIY,WORK(L1),WORK(L2) )

```

C

```

IF (IFNEWH.EQ. 0) GO TO 250
NTDACC = 34
NROW = NODES + 4
MAXSEC = 10
MAXPR = 10

```

C

```

11 NAA + NTP
12 I1 + NTP
13 I2 + NTP
14 I3 + NODES
15 I4 + NODES
16 I5 + NODES
17 I6 + NTP
18 I7 + NROW
19 I8 + MAXPR
20 I9 + MAXSEC
21 I10 + MAXPR*MAXSEC*2
22 I11 + NROW*NROW
23 I12 + NROW
CALL NEWH (NTI, NOUT, NTDACF, MASTH4, MASTH, MASTDH, NTP, NSYM,
1 NASYM, NROW, MODES, NBARAY, NP, NCARAY, YS, ZS, WORK(I1),
2 XIC, X, WORK(I2), M, DHDX, WORK(I3), WORK(I4), WORK(I5),
3 ELXI, ELVI, ELZI, WORK(I6), PHIN, WORK(I7), WORK(I8),
4 WORK(I9), WORK(I10), WORK(I11), WORK(I12), NODES, MDOFF,
5 PHI, MDOFF, NODES)
NAA = IFNEWH
250 CONTINUE
IF (NGUST.EQ.0) GO TO 270
WRITE (NOUT,120)
252 READ (NTI,30) NN, XCOS, YCOS, ZCOS
IF (NN.LE.0) GO TO 253
CR(1,NN) = XCOS
CR(2,NN) = YCOS
CR(3,NN) = ZCOS
GO TO 252
253 CONTINUE
DO 260 NR = 1, NG
WRITE (NOUT,130) NR, (CR(MR,NR), MR=1,3)
260 CONTINUE
C
270 CONTINUE
NMSYM = NSYM + NG
NMASYM = NASYM + NG
NMOTOT = NMSYM + NMASYM
CALL WRAERO (4HCR,0,CR,NOUT,NER)
CALL WRAERO (4HCR,0,FREQ,NOUT,NER)
REWIND NTGF
L1 = NAA
C *** START OF FREQUENCY LOOP
C
DO 370 JKR = 1, NK
REWIND MASTCP
REWIND MASTFZ
REWIND MASTFY
KR = FREQ(JKR)
IF (JKR.GT. NKO) GO TO 362
IF (IOPT.EQ.4) GO TO 360
IF (IOPT.EQ.2) GO TO 280

```

```

C
L2 = L1 + 2*NTOT
L3 = L2 + 2*NTOT
L4 = L3 + 2*NTOT
NAA = L4 + 2*NTOT

CALL GEND (IPR3,NTAPE,NTOT,IOPT,MASTDT,WORK(L1),
1 WORK(L1),WORK(L2),WORK(L3),WORK(L4)
2 , IFLA,NBEA,NT12,NBARAY,NCARAY,YB,ZB,ARB,AVR,RIA,XLE
3 , XTE,TH1A,TH2A,X,CG,EE,SG,YS,ZS,XIC,YIN,ZIN,DELX
4 , XLAM,SCALER)

C
IF (JKR.EQ.NK.AND. IOPT.EQ.1) END FILE MASTDT
280 CONTINUE
IF (NB.EQ.0) GO TO 290
REWIND MASTH
REWIND MASTDH
READ (MASTH) NSYMB,NASYMB
READ (MASTDH)

C
CALL SB( NMT , NSYMB , NASYMB, NOUT, MASTH, MASTDH,
1 IPR2, WORK(L1), H, DT, DTA
2 , NBEA, NSBEA, NCARAY, NSARAY, AO, YB, ZB, ARB
3 , AOP, XIS1, XIS2, X, CG, SG, YS, ZS, CR, NG )

C
NSBMIN = NSBEIO
NSBET2 = 2*NSBEIO
NTSBE = NTZS + NTYS
LENGTH = NTOT + NTSBE
L2 = L1 + 4*LENGTH

C
CALL BFSMAT(LENGTH, NE , NB , NP , NTP ,
1 NBFM , NBFMA , FMACH , REFC , KR , YB ,
2 ZB , YS , ZS , X , DELX , EE ,
3 SG , CG , ARB , RIA , AO , XIS1 ,
4 AVR , NSARAY , NCARAY , NSBEA , NBEA(1,1) ,
5 NAS , NASB , WORK(L1), WORK(L2), SCALER )
CONTINUE
REWIND NWT
REWIND NWT
L2 = L1 + 2*NTOT
L3 = L2 + 2*NTOT

C
CALL WANCWT (IPR1,LINES,NOUT,NTOT,NTP,NB,NDW,MASTH,
1 MASTDH,NWT,NWT,
2 KR, H, DHDX, WORK(L1), WORK(L2), WORK(L3)
REFC)

C
CALL GUST( NWT , NWT , NWT , NTP , NB, NBARAY, NCARAY,
1 NBEA, FMACH, KR, REFC, X, YS, ZS, SG, CR, NG,
2 IPR1, LINES, NOUT, WORK(L1), WORK(L2), NDW, NMT, NASYM, DT )

C
NMSYM = NSYM + NG
NMASYM = NASYM + NG

```

310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361

```

IF (NASYM -EQ, 0) NMAASYM = 0
NMDTOT = NMSYM + NMAASYM
NWTAP = NWT
NMODE = NMSYM
IGO = 1
NFTP = NBFM
NMODEB = NSYM
CONTINUE = NTOT * NMODE
NRHSTO = NWTAP
REWIND NI
JX2 = L1 - 1
DO 310 JJ = 1, NMODE
JX1 = JX2 + 1
JX2 = JX2 + 2*NTOT
CALL RWREC(IREAD, NWTAP, WORK(JX1), NTOT, NUMBR, IZERO)
310 CONTINUE
C
IF (IOPT, NE, 2) REWIND ITAPE
DO 340 IX = 1, NTOT
JM = 0
READ (ITAPE) (DT(II), II=1, NTOT), (DTA(II), II=1, NTOT)
CALL RHSTO(NTOT, NMODE, JM, IX, MK1, MK2, NRHSTO, RHS,
1 WORK(L1))
C
GO TO (320, 330), IGO
CONTINUE
WRITE (NI) (DT(IK), IK=1, NTOT), (RHS(JK), JK=1, NMODE)
GO TO 340
330 WRITE (NI) (DTA(IK), IK=1, NTOT), (RHS(JK), JK=1, NMODE)
340 CONTINUE
C
REWIND NI
REWIND NW
CALL ZEROUT (WORK(L1), NWORK, 0, 0)
NPM = NTOT + NMODE
NWORK = 2*MAXO(3*NPM, NTOT*NMODE)
WRITE (NOUT, 80)
CALL SOLVIT(WORK(L1), WORK(L1), NTOT, NMODE, NWORK,
1 NI, NM, NO, NW, IPR3)
REWIND NW
RWREC(ICOPY, NW, WORK(L1), NTOT, NMODE, MASTCP)
CALL (NB -EQ, 0) GO TO 342
L5 = L1 + 2*LENGTH
LWORK = LENGTH
CALL MATMUL(NW, NPSTAP, NFTP, NEWBFZ, NEWBFY, LWORK, NTSBE,
1 NSBETO, LENGTH, NTOT, NMODE, NMODEB, IPR3, KR,
2 WORK(L1), WORK(L1), WORK(L5), WORK(L5))
REWIND NEWBFZ
REWIND NEWBFY
CALL RWREC(ICOPY, NEWBFZ, WORK(L1), NSBETO, NMODE, MASTFZ)
CALL RWREC(ICOPY, NEWBFY, WORK(L1), NSBETO, NMODE, MASTFY)

```

SDLM 362
SDLM 363
SDLM 364
SDLM 365
SDLM 366
SDLM 367
SDLM 368
SDLM 369
SDLM 370
SDLM 371
SDLM 372
SDLM 373
SDLM 374
SDLM 375
SDLM 376
SDLM 377
SDLM 378
SDLM 379
SDLM 380
SDLM 381
SDLM 382
SDLM 383
SDLM 384
SDLM 385
SDLM 386
SDLM 387
SDLM 388
SDLM 389
SDLM 390
SDLM 391
SDLM 392
SDLM 393
SDLM 394
SDLM 395
SDLM 396
SDLM 397
SDLM 398
SDLM 399
SDLM 400
SDLM 401
SDLM 402
SDLM 403
SDLM 404
SDLM 405
SDLM 406
SDLM 407
SDLM 408
SDLM 409
SDLM 410
SDLM 411
SDLM 412
SDLM 413

```

342 GO TO (350,360), IGO
350 CONTINUE
  IF (NMASYM.EQ.0) GO TO 360
  IGO = 2
  NWTAP = NMTA
  NMODE = NMASYM
  NFTP = NBFMA
  NMODEB = NASYM
  GO TO 300
C
360 CONTINUE
  GO TO 368
362 IF (FMACH.GT. 0.01) GO TO 366
C
364 WRITE (6,364)
  FORMAT ( ' /// 68H *** PISTON THEORY NOT APPLICABLE AT THIS MACH
1NUMBER *** STOP
STOP
C
366 CONTINUE
  JRUN = JRUN + 1
  IA1 = L1
  IA2 = IA1 + 2*MAXO(NTP, NSBETO)
  IA3 = IA2 + 2*MAXO(NTP, NSBETO)
  IA4 = IA3 + 2*NSBETO
C
  CALL
1 PISTON( NB , NTP , NSBETO, KR , REEC , FMACH,
2 MASTH , MASTDH, MASTH4, MASTCP, MASTFY, ITAPE,
3 NBARAY , NCARAY, NSBEA , YB , AO , AOP ,
4 XIS1 , XIS2 , CG, SG, EE, YS, ZS, H , DHDH , DT ,
5 XC ,
6 WORK(IA1), WORK(IA2), WORK(IA3) ,
  WORK(IA4), WORK(IA4) , NG, CR )
C
  MASTH4 = ITAPE
  MTOT = NTP
368 CONTINUE
  REWIND MASTCP
  IF (NB.NE.0) REWIND MASTFZ
C
  NSMAX = MAXO(NSRIP, NSBETO)
  NSMAX = MAXO(NSMAX, NMDTOT)
  L2 = L1 + 2*NTMAX
  L3 = L2 + 2*NTMAX
C
  CALL
1 GENF(NMODES,MTOT,NTP,NB, YB , NSBETO,IPR2,
2 LINES, NOUT, NDZ , NDY , MASTH , MASTH4, NTAERO,
3 NTGF, MASTCP, MASTFZ, MASTFY, JKR, H, DHDH, NASYM,
  DELA, XIS1, XIS2, NSBEA, WORK(L1), WORK(L2), WORK(L3), NG )
C

```

```

L2 2*NTOT
L3 2*NSMAX
L4 2*NSMAX
L5 2*NSMAX
L6 2*NSMAX
L7 2*NSMAX
L8 2*NSMAX
L9 2*NSMAX
L10 2*NSMAX
L11 2*NSMAX
L12 2*NSMAX
CALL MSTRIP
1 AERO(MFIX1,MFIX2,NG,NMSYM,MTOT,NOUT,MASTCP,MASTFZ,
2 MASTFY,NSMAX,REFC,WORK(L1),WORK(L2),WORK(L3),
3 WORK(L4),WORK(L5),WORK(L6),
4 ISSTR,NSBEA,NBARAY,NCARAY,YB,ZB,XIS1,XIS2,CG,CS,EE,SG,YS,ZS
5 ,XIC,XIJ,DELX,COORD
6 WORK(L7),WORK(L8),WORK(L9),WORK(L10),WORK(L11),WORK(L12)
C 370 CONTINUE
C *** END OF FREQUENCY LOOP
C RETURN
C END

```

```

SDLM 466
SDLM 467
SDLM 468
SDLM 469
SDLM 470
SDLM 471
SDLM 472
SDLM 473
SDLM 474
SDLM 475
SDLM 476
SDLM 477
SDLM 478
SDLM 479
SDLM 480
SDLM 481
SDLM 482
SDLM 483
SDLM 484
SDLM 485
SDLM 486
SDLM 487
SDLM 488
SDLM 489

```


SNPDF 54
SNPDF 55
SNPDF 56
SNPDF 57
SNPDF 58
SNPDF 59
SNPDF 60
SNPDF 61
SNPDF 62
SNPDF 63
SNPDF 64
SNPDF 65
SNPDF 66
SNPDF 67
SNPDF 68
SNPDF 69
SNPDF 70
SNPDF 71
SNPDF 72
SNPDF 73
SNPDF 74
SNPDF 75
SNPDF 76
SNPDF 77
SNPDF 78
SNPDF 79
SNPDF 80
SNPDF 81
SNPDF 82
SNPDF 83
SNPDF 84
SNPDF 85
SNPDF 86
SNPDF 87
SNPDF 88
SNPDF 89
SNPDF 90
SNPDF 91
SNPDF 92

```

30 IF (CAB*CBH)40,50,50
40 CACB = 0.
   GO TO 60
50 CACB = 0.5*ABS((1./RIMAG**2)-(1./ROMAG**2))
60 CONTINUE
   VBZ = CACB * (DBX*CLSGS - DBZ*SLB)
   VBY = CACB * (DBY*SLB - DBX*CLCGS)
   *** TEST TO SEE IF THE RECEIVING POINT LIES ON OR NEAR THE
   C INBOARD TRAILING VORTEX -- IF SO, THE CONTRIBUTION OF THE
   C INBOARD TRAILING VORTEX IS SET TO ZERO
   IF (D12.GT.TFST2) GO TO 62
   VIY = 0.0
   VIZ = 0.0
   GO TO 64
62 CONTINUE
   ONECBI = (1.0 - CBI)/D12
   VIY = ONECBI*RIZ
   VIZ = -ONECBI*RIY
64 CONTINUE
   *** TEST TO SEE IF THE RECEIVING POINT LIES ON OR NEAR THE
   C OUTBOARD TRAILING VORTEX -- IF SO, THE CONTRIBUTION OF THE
   C OUTBOARD TRAILING VORTEX IS SET TO ZERO
   IF (D02.GT.TFST2) GO TO 66
   VOY = 0.0
   VOZ = 0.0
   GO TO 68
66 CONTINUE
   CAONE = (1.0 + CA0)/D02
   VOY = -CAONE*ROZ
   VOZ = CAONE*ROY
68 CONTINUE
   VX = DBZ*CLCGS - DBY*CLSGS
   VY = VBY + VIY + VOY
   VZ = VBZ + VIZ + VOZ
   WW = VY*SGR - VZ*CGR
   CLJ = WW*CAVE / 25.132741
70 CONTINUE
   RETURN
   END

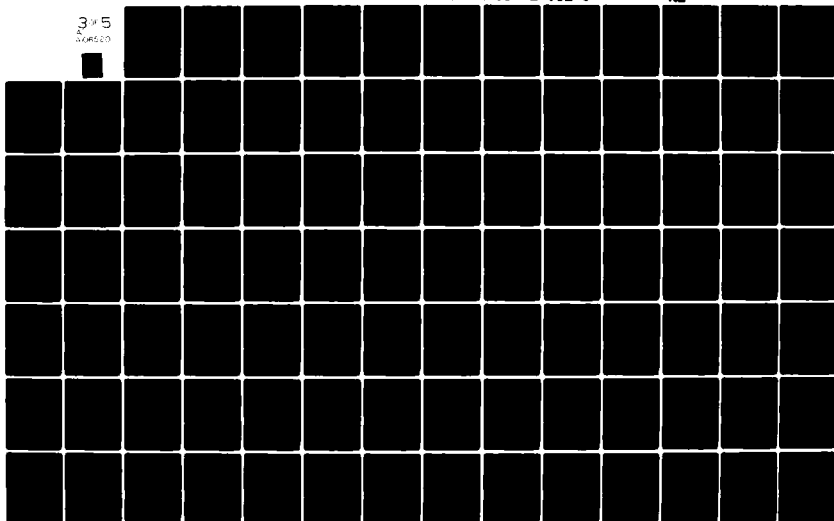
```

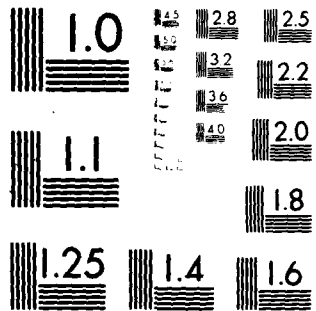

AD-A106 520

DOUGLAS AIRCRAFT CO. LONG BEACH CA F/O 18/3
NUCLEAR BLAST RESPONSE COMPUTER PROGRAM, VOLUME III. PROGRAM LI--ETC(U)
AUG 81 J A MCOREY, H H CROXEN, T P KALMAN DHA001-75-C-0216
AFWL-TR-81-32-VOL-3 NL

UNCLASSIFIED

3 of 5
AD-A106 520





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

C - - IF (LAST) K = NN
C - - READ *K* ROWS OF THE AUGMENTED 'A' MATRIX
C
30 NT = 0
DO 40 IB = 1, K
NS = NT + 1
NT = NT + NEL
40 READ (NIN) (A(IO), IO = NS, NT)
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROW
C
IF (K .EQ. 1) GO TO 90
C - - *K* IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION
C
NELP1 = NEL + 1
NS = - NEL
NELP2 = NELP1 + 1
C - - FORM THE 'TRAPEZOIDAL' ARRAY (8)
C
DO 50 IB = 2, K
NP = NELP2 - IB
NS = NS + NELP1
NT = NS
DO 50 IO = IB, K
NT = NT + NEL
MN = NT
NB = NS
A(NT) = -A(NT) / A(NS)
TEMPR = REAL(A(NT))
DENOM = - ( REAL(A(NS)) ** 2 + AIMAG(A(NS)) ** 2 )
RA(1,NT) = ( TEMPR(A(NS)) * REAL(A(NT)) +
* AIMAG(A(NS))) / DENOM
RA(2,NT) = ( AIMAG(A(NT)) * REAL(A(NS)) - TEMPR *
* AIMAG(A(NS))) / DENOM
C
DO 50 NF = 2, NP
MN = MN + 1
NB = NB + 1
C
50 A(MN) = A(MN) + A(NT) * A(NB)
RA(1,MN) = RA(1,MN) + REAL(A(NT)) * REAL(A(NB)) -
* AIMAG(A(NT)) * AIMAG(A(NB))
C
50 RA(2,MN) = RA(2,MN) + REAL(A(NT)) * AIMAG(A(NB)) +
* AIMAG(A(NT)) * REAL(A(NB))
C

```

SOLVIT54
SOLVIT55
SOLVIT56
SOLVIT57
SOLVIT58
SOLVIT59
SOLVIT60
SOLVIT61
SOLVIT62
SOLVIT63
SOLVIT64
SOLVIT65
SOLVIT66
SOLVIT67
SOLVIT68
SOLVIT69
SOLVIT70
SOLVIT71
SOLVIT72
SOLVIT73
SOLVIT74
SOLVIT75
SOLVIT76
SOLVIT77
SOLVIT78
SOLVIT79
SOLVIT80
SOLVIT81
SOLVIT82
SOLVIT83
SOLVIT84
SOLVIT85
SOLVIT86
SOLVIT87
SOLVIT88
SOLVIT89
SOLVIT90
SOLVIT91
SOLVIT92
SOLVIT93
SOLVIT94
SOLVIT95
SOLVIT96
SOLVIT97
SOLVIT98
SOLVIT99
SOLVIT100
SOLVIT101
SOLVIT102
SOLVIT103
SOLVIT104
SOLVIT105

SOLV1106
SOLV1107
SOLV1108
SOLV1109
SOLV1110
SOLV1111
SOLV1112
SOLV1113
SOLV1114
SOLV1115
SOLV1116
SOLV1117
SOLV1118
SOLV1119
SOLV1120
SOLV1121
SOLV1122
SOLV1123
SOLV1124
SOLV1125
SOLV1126
SOLV1127
SOLV1128
SOLV1129
SOLV1130
SOLV1131
SOLV1132
SOLV1133
SOLV1134
SOLV1135
SOLV1136
SOLV1137
SOLV1138
SOLV1139
SOLV1140
SOLV1141
SOLV1142
SOLV1143
SOLV1144
SOLV1145
SOLV1146
SOLV1147
SOLV1148
SOLV1149
SOLV1150
SOLV1151
SOLV1152
SOLV1153
SOLV1154
SOLV1155
SOLV1156
SOLV1157

```

IF (LAST) GO TO 90
C -- WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE
C
C
NT = 0
NP = NEL
NS = - NEL
DO 60 IO = 1, K
NS = NS + NELP1
NT = NT + NEL
WRITE (MT) NP, (A(IB), IB = NS, NT)
60 NP = NP - 1
NP = NP - M
NS = KORE - NEL + 1
C -- READ ANOTHER ROW
C
C
DO 80 IO = 1, NP
READ (NIN) (A(IB), IB = NS, KORE)
C -- MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY
C
C
NT = 1
MN = NS
DO 70 IB = 1, K
NB = NT
NF = MN + 1
A(MN) = -A(MN) / A(NT)
TEMPR = REAL(A(MN))
DENOM = - (REAL(A(NT)) ** 2 + AIMAG(A(NT)) ** 2)
RA(1,MN) = (TEMPR * REAL(A(NT)) + AIMAG(A(MN)) *
* AIMAG(A(NT))) / DENOM
RA(2,MN) = (AIMAG(A(MN)) * REAL(A(NT)) - TEMPR *
* AIMAG(A(NT))) / DENOM
C
DO 65 NN = NF, KORE
NB = NB + 1
C
C
65 A(NN) = A(NN) + A(MN) * A(NB)
RA(1,NN) = RA(1,NN) + REAL(A(MN)) * REAL(A(NB)) -
* AIMAG(A(MN)) * AIMAG(A(NB))
C
65 RA(2,NN) = RA(2,NN) + REAL(A(MN)) * AIMAG(A(NB)) +
* AIMAG(A(MN)) * REAL(A(NB))
C
MN = NF
70 NT = NT + NELP1
C -- WRITE THE MODIFIED ROW ON TAPE
C
C

```

SOLVI158
SOLVI159
SOLVI160
SOLVI161
SOLVI162
SOLVI163
SOLVI164
SOLVI165
SOLVI166
SOLVI167
SOLVI168
SOLVI169
SOLVI170
SOLVI171
SOLVI172
SOLVI173
SOLVI174
SOLVI175
SOLVI176
SOLVI177
SOLVI178
SOLVI179
SOLVI180
SOLVI181
SOLVI182
SOLVI183
SOLVI184
SOLVI185
SOLVI186
SOLVI187
SOLVI188
SOLVI189
SOLVI190
SOLVI191
SOLVI192
SOLVI193
SOLVI194
SOLVI195
SOLVI196
SOLVI197
SOLVI198
SOLVI199
SOLVI200
SOLVI201
SOLVI202
SOLVI203
SOLVI204
SOLVI205
SOLVI206
SOLVI207
SOLVI208
SOLVI209

```

C      80 WRITE (NOUT)      (A(NT), NT = MN, KORE)
      REWIND NOUT
      REWIND NIN
C      -- SWITCH THE TAPES
C      NT = NIN
C      NIN = NOUT
C      NOUT = NT
C      -- RE-CALCULATE ROW LENGTH AND LOOP BACK
C      NEL = NEL - K
C      NN = NEL - M
C      GO TO 10
C      -- REWIND ALL TAPES
C      90 REWIND MT
C      REWIND NIN
C      REWIND NOUT
C      -- CONDENSE THE MATRIX
C      NN = NEL
C      NL = NEL + 1
C      IF (K.EQ. 1) GO TO 105
C      NS = 1
C      NT = NEL
C      DO 100 IR = 2, K
C      NS = NS + NEL*PI
C      NT = NT + NEL
C      DO 100 IO = NS, NT
C      A(NI) = A(IO)
C      NI = IO + 1
C      NI = NI - K * M + 1
C      100 NL = NI
C      105 NI = NI - K * M + 1
C      -- THERE, NOW WE CAN START THE BACK-SOLUTION
C      * NOTE: THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(NI)
C      NREM = N
C      NEL = NPM
C      LAST = K.EQ. N
C      NPASS = 0
C      -- SOLVE FOR THE ANSWERS CORRESPONDING TO *K* ROWS
C      110 KMI = K - 1
C      KPI = K + 1
C      NS = NL - MPI
C      NPASS = NPASS + 1

```

```

C
C
C
DO 130 MN = 1, M
NF = NS + MN
A(NF) = A(NF) / A(NS)
TEMPR = REAL(A(NF))
DENOM = REAL(A(NS)) ** 2 + AIMAG(A(NS)) ** 2
RA(1,NF) = (TEMPR * REAL(A(NS))) / DENOM * AIMAG(A(NF)) *
*
RA(2,NF) = (AIMAG(A(NF)) * REAL(A(NS)) - TEMPR *
* AIMAG(A(NS))) / DENOM
C
NT = NS
IF (KMI.EQ.0) GO TO 130
DO 125 IB = 1, KMI
NF = NF - IB - M
NT = NT - MPI - IB
SUM = 0.0
SUMR = 0.0
SUMI = 0.0
C
NP = NF
N2 = MPI + IB
DO 120 IO = 1, IB
NN = NT + IO
NP = NP + N2 - IO
C 120 SUM = SUM + A(NN) * A(NP)
C
SUMR = SUMR + REAL(A(NN)) * REAL(A(NP)) - AIMAG(A(NN)) *
* AIMAG(A(NP))
C 120 SUMI = SUMI + REAL(A(NN)) * AIMAG(A(NP)) + AIMAG(A(NN)) *
* REAL(A(NP))
C
C 125 A(NF) = (A(NF) - SUM) / A(NT)
DENOM = REAL(A(NT)) ** 2 + AIMAG(A(NT)) ** 2
TEMPR = REAL(A(NF)) - SUMR
TEMPI = AIMAG(A(NF)) - SUMI
C
RA(1,NF) = (TEMPR * REAL(A(NT)) + TEMPI * AIMAG(A(NT))) / DENOM
125 RA(2,NF) = (TEMPI * REAL(A(NT)) - TEMPR * AIMAG(A(NT))) / DENOM
C 130 CONTINUE
C
C -- MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(NI)
C
NI = KORE + 1

```

SOLVI210
 SOLVI211
 SOLVI212
 SOLVI213
 SOLVI214
 SOLVI215
 SOLVI216
 SOLVI217
 SOLVI218
 SOLVI219
 SOLVI220
 SOLVI221
 SOLVI222
 SOLVI223
 SOLVI224
 SOLVI225
 SOLVI226
 SOLVI227
 SOLVI228
 SOLVI229
 SOLVI230
 SOLVI231
 SOLVI232
 SOLVI233
 SOLVI234
 SOLVI235
 SOLVI236
 SOLVI237
 SOLVI238
 SOLVI239
 SOLVI240
 SOLVI241
 SOLVI242
 SOLVI243
 SOLVI244
 SOLVI245
 SOLVI246
 SOLVI247
 SOLVI248
 SOLVI249
 SOLVI250
 SOLVI251
 SOLVI252
 SOLVI253
 SOLVI254
 SOLVI255
 SOLVI256
 SOLVI257
 SOLVI258
 SOLVI259
 SOLVI260
 SOLVI261

```

DO 140 NN = 1, K
DO 135 MN = 1, M
NL = NL - 1
NI = NI - 1
135 A(NI) = A(NL)
140 NL = NL - NN
CC
-- WRITE THE SOLUTIONS ON TAPE
WRITE (NIN) K
NS = NI - 1
DO 145 MN = 1, M
NT = NS + MN
145 WRITE (NIN) (A(IO), IO = NT, KORE, M)
CC
-- TEST IF THIS IS THE LAST PASS
IF (LAST) GO TO 200
CC
-- WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
THE SOLUTIONS OBTAINED SO FAR (EQ 21)
* * NOTE--LOCATIONS A(1) TO A(NI-1) ARE NOW FREE TO USE
CC
-- CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'
NELOLD = NEL
KOLD = K
NEL = NEL - K
NREM = NREM - K
CC
-- NOW APPLY THE INCREDIBLE FORMULA FOR THE NEW 'K'
K = (-4 * M - 1) / 2 + IFIX(SQRT(0.25 + FLOAT((4 * M + 2) * M +
1 2 * (KORE - NELOLD))))
NROW = NREM - K + 1
IF (K .LT. NREM) GO TO 150
LAST = .TRUE.
NROW = 1
K = NREM
150 NS = 1
NT = NELOLD + 1
CC
-- READ IN THE ROWS TO BE MODIFIED
DO 190 IB = 1, NREM
NT = NT - 1
IF (IB .LE. NROW) GO TO 160
NS = NS + NN
NT = NT + NN
160 READ (MT) NN, (A(IO), IO = NS, NT)
NP = NI - 1
NF = NT - M - KMI

```

SOLVI262
SOLVI263
SOLVI264
SOLVI265
SOLVI266
SOLVI267
SOLVI268
SOLVI269
SOLVI270
SOLVI271
SOLVI272
SOLVI273
SOLVI274
SOLVI275
SOLVI276
SOLVI277
SOLVI278
SOLVI279
SOLVI280
SOLVI281
SOLVI282
SOLVI283
SOLVI284
SOLVI285
SOLVI286
SOLVI287
SOLVI288
SOLVI289
SOLVI290
SOLVI291
SOLVI292
SOLVI293
SOLVI294
SOLVI295
SOLVI296
SOLVI297
SOLVI298
SOLVI299
SOLVI300
SOLVI301
SOLVI302
SOLVI303
SOLVI304
SOLVI305
SOLVI306
SOLVI307
SOLVI308
SOLVI309
SOLVI310
SOLVI311
SOLVI312
SOLVI313

```

NN = NN - KOLD
DO 170 MN = 1, M
N2 = NF
NA = NP + MN
NB = NA
SUM = 0.0
SUMR = 0.0
SUMI = 0.0
DO 165 IO = 1, KOLD
SUM = SUM + A(N2) * A(NA)
SUMR = SUMR + REAL(A(N2)) * REAL(A(NA)) - AIMAG(A(N2)) *
* AIMAG(A(NA))
SUMI = SUMI + REAL(A(N2)) * AIMAG(A(NA)) + AIMAG(A(N2)) *
* REAL(A(NA))
N2 = N2 + 1
165 NA = NA + M
N2 = N2 + MN - 1
C 170 A(N2) = A(N2) - SUM
RA(1,N2) = RA(1,N2) - SUMR
170 RA(2,N2) = RA(2,N2) - SUMI
C -- WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW
NL = NT - M + 1
IF (IB .GE. NROW) GO TO 175
NF = NL - KPI
WRITE (NOUT) NN, (A(IO), IO = NS, NF), (A(IO), IO = NL, NT)
GO TO 190
175 NF = NL - KOLD
DO 180 MN = NL, NT
A(NF) = A(MN)
180 NF = NF + 1
190 CONTINUE
REWIND MT
REWIND NOUT
C -- SWITCH THE TAPES
NT = MT
MT = NOUT
NOUT = NT
C -- LOOP BACK THRU THE SOLUTION

```

SOLVI314
SOLVI315
SOLVI316
SOLVI317
SOLVI318
SOLVI319
SOLVI320
SOLVI321
SOLVI322
SOLVI323
SOLVI324
SOLVI325
SOLVI326
SOLVI327
SOLVI328
SOLVI329
SOLVI330
SOLVI331
SOLVI332
SOLVI333
SOLVI334
SOLVI335
SOLVI336
SOLVI337
SOLVI338
SOLVI339
SOLVI340
SOLVI341
SOLVI342
SOLVI343
SOLVI344
SOLVI345
SOLVI346
SOLVI347
SOLVI348
SOLVI349
SOLVI350
SOLVI351
SOLVI352
SOLVI353
SOLVI354
SOLVI355
SOLVI356
SOLVI357
SOLVI358
SOLVI359
SOLVI360
SOLVI361
SOLVI362
SOLVI363
SOLVI364
SOLVI365

```

NL = NF
GO TO 110
C -- START TO WRAP IT UP
C 200 END FILE NIN
REWIND NIN
N2 = N
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KURE) ARE FREE
C
DO 220 IB = 1, NPASS
READ (NIN) K
N1 = N2 - K + 1
NS = N1
NT = N2
C -- READ IN THE SOLUTIONS
C
DO 210 IC = 1, M
READ (NIN) (A(NN), NN = NS, NT)
NT = NT + N
210 NS = NS + N
220 N2 = N1 - 1
C -- WRITE THE SOLUTIONS ON TAPE
C
NT = 0
LINES = 0
DO 230 IO = 1, M
NS = NT + 1
NT = NT + N
IF (NPR1.EQ.0) GO TO 230
LINES = LINES + ND/3 + 1
WRITE (6,3) IO
WRITE (6,2) (A(NN), NN = NS, NT)
IF (LINES.LT.25) GO TO 230
WRITE (6,4)
LINES = 0
230 WRITE (NW) (A(NN), NN = NS, NT)
RETURN
END

```

SOLVI366
 SOLVI367
 SOLVI368
 SOLVI369
 SOLVI370
 SOLVI371
 SOLVI372
 SOLVI373
 SOLVI374
 SOLVI375
 SOLVI376
 SOLVI377
 SOLVI378
 SOLVI379
 SOLVI380
 SOLVI381
 SOLVI382
 SOLVI383
 SOLVI384
 SOLVI385
 SOLVI386
 SOLVI387
 SOLVI388
 SOLVI389
 SOLVI390
 SOLVI391
 SOLVI392
 SOLVI393
 SOLVI394
 SOLVI395
 SOLVI396
 SOLVI397
 SOLVI398
 SOLVI399
 SOLVI400
 SOLVI401
 SOLVI402
 SOLVI403
 SOLVI404
 SOLVI405
 SOLVI406
 SOLVI407
 SOLVI408

```

SUBROUTINE SORT(IF1,IF2,NSUP,INSUP,I1,I2,NXQ,LMAX,X1L,X2L,X3L,YL,
* XP,YP,H,DHDX,I5P,NSBP,NBCUM,NPCUM,NTPH,NTPDH,
3 NTPSH,KPRINT,NTP6,KCUM1,KCUM2,KCUM3,NTOT,ISORT,NODE,
* SUPRH,SUPRDH)
* DIMENSION INSUP(200),X1L(LMAX),X2L(LMAX),X3L(LMAX),YL(LMAX),
* XP(LMAX),YP(LMAX),YPL(LMAX),NBCUM(50),NPCUM(200)
* DIMENSION H(1),DHDX(1),NODE(1),SUPRH(1),SUPRDH(1)
*** ISORT IS A FLAG THAT ACTIVATES EITHER SEGMENT 1 (ISORT=1)
OR SEGMENT 2 (ISORT=2) OF THE SUBROUTINE
GO TO (70,270),ISORT
70 CONTINUE
*** SEGMENT 1 EXTRACTS THE XP- AND YP- ARRAYS FOR THE
CURRENT SUPERPANEL (BODY) FROM THE COMPLETE GEOMETRY
ARRAYS SUPPLIED BY SUBROUTINE DATA
DO 90 I=1,LMAX
XP(I)=0.0
YP(I)=0.0
90 CONTINUE
DO 200 J=1,NSUP
IF (IF1.EQ.0) GO TO 110
NBODY=INSUP(J)
NBML=NBODY-1
IF (NBML.EQ.0) GO TO 95
L1=NBCUM(NBML)+1
GO TO 100
95 L1=1
100 L2=NBCUM(NBODY)
110 GO TO 130
CONTINUE
INSUP(J)=NOPAN-1
NPM1=0
IF (NPM1.EQ.0) GO TO 115
L1=1
GO TO 120
115 L1=1
120 L2=1
130 CONTINUE
INSUP(J)=NPCUM(NOPAN)
DO 180 L=L1,L2
IF (IF1.EQ.0) GO TO 170
XP(I)=X3L(L)
YP(I)=0.0
GO TO 180
170 CONTINUE
YP(I)=YL(L)

```

```

IF (IF2 .EQ. 1) GO TO 175
XP(I) = X1L(L)
GO TO 180
175 XP(I) = X2L(L)
180 CONTINUE
200 CONTINUE = I
I2
RETURN
C
C
C 270 CONTINUE
C
C *** SEGMENT 2 EXPANDS THE H- AND DHDX- COLUMNS GENERATED BY
C SUBROUTINE SPL2 FOR ONE SUPERPANEL(BODY) INTO FULL
C COLUMNS -SUPRH- AND -SUPRDH- , AND SAVES THEM ON
C TAPE UNITS NTPSH AND NTPSDH , RESPECTIVELY.
C
DO 500 K=1,NXQ
READ (NTPH) (H(I), I=11,12)
IF (IF1 .EQ. 0 .AND. IF2 .EQ. 1) GO TO 275
KCUM1 = KCUM1 + 1
IF (IF2 .NE. 0) GO TO 280
READ (NTPDH) (DHDX(I), I=11,12)
KCUM2 = KCUM2 + 1
GO TO 280
275 CONTINUE = KCUM3 + 1
280 CONTINUE
DO 290 L=1,LMAX
SUPRH(L) = 0.0
290 SUPRDH(L) = 0.0
IH = 0
L1 = 1
L2 = 1
DO 400 J=1,NSUP
IF (IF1 .EQ. 0) GO TO 310
NBODY = INSUP(J)
NBMI = NBODY - 1
IF (NBMI .EQ. 0) GO TO 305
L1 TO 300 = NRCUM(NBMI) + 1
305 L1 = 1
300 L2 = NRCUM(NBODY)
310 CONTINUE = INSUP(J)
NOPAN = NOPAN - 1
IF (NPMI .EQ. 0) GO TO 315
L1 = NPCUM(NPMI) + 1
GO TO 320
315 L1 = 1

```

SORT 54
 SORT 55
 SORT 56
 SORT 57
 SORT 58
 SORT 59
 SORT 60
 SORT 61
 SORT 62
 SORT 63
 SORT 64
 SORT 65
 SORT 66
 SORT 67
 SORT 68
 SORT 69
 SORT 70
 SORT 71
 SORT 72
 SORT 73
 SORT 74
 SORT 75
 SORT 76
 SORT 77
 SORT 78
 SORT 79
 SORT 80
 SORT 81
 SORT 82
 SORT 83
 SORT 84
 SORT 85
 SORT 86
 SORT 87
 SORT 88
 SORT 89
 SORT 90
 SORT 91
 SORT 92
 SORT 93
 SORT 94
 SORT 95
 SORT 96
 SORT 97
 SORT 98
 SORT 99
 SORT 100
 SORT 101
 SORT 102
 SORT 103
 SORT 104
 SORT 105

SORT 106
 SORT 107
 SORT 108
 SORT 109
 SORT 110
 SORT 111
 SORT 112
 SORT 113
 SORT 114
 SORT 115
 SORT 116
 SORT 117
 SORT 118
 SORT 119
 SORT 120

```

320 L2      = NPCUM(NOPAN)
330 CONTINUE
DO 380 L=L1,L2
  IH = IH + 1
  SUPRH(L) = H(IH)
  IF (IF2 .NE. 0) GO TO 380
  SUPROH(L) = DHDX(IH)
380 CONTINUE
400 CONTINUE
WRITE (NTPSH) NODE(K), NTOT, (SUPRH (I), I=1, NTOT)
IF (IF2 .NE. 0) GO TO 500
WRITE (NTPSH) NODE(K), NTOT, (SUPRDH(I), I=1, NTOT)
500 CONTINUE
RETURN
END
  
```

```

SUBROUTINE SPLINE (MASTH,MASTH4,MASTH4,NMODEB,NMODE,
1 ELXI, ELYI, ELZI, WORK
2 , NSBEA, NBARAY, NCARAY, NSAKAY, XIS1, XIS2, X, CG, SG, YS, ZS
3 , XIC, XIJ, PHI, NCARAY, PHIZ, PHIZ, MOOF, ANODE, NMODES, NDOFB, IFNEWH
4 , NBCUM, INSUP, NODE, XIJ, ETAJ, CSG
5 , X3L, YL, XP, YP, H, DHDX )
  DIMENSION ELXI(1), ELYI(1), ELZI(1)
  DIMENSION WORK(1)
  DIMENSION NSBEA(1), NBARAY(1), NCARAY(1), NSARAY(1), XIS1(1)
  DIMENSION XIS2(1), X(1), CG(1), SG(1), YS(1), ZS(1), XIC(1), XIJ(1)
  *** SURFACE SPLINE PROGRAM

COMMON /AROCOM/ NTI, MODES
X , NP, MSTRIP, NSMAX, NCMAX, NTOTAL, NB, MSBE, MBE
Y , ND, NE, NBY, NBZ, NTO, NTP, NTY, NTZ
1 , NTYS, NTZS, MAXGR, MAXSTR, NSBEU, NSTRIP, KR, XM, REFA, REFC
2 , REFS, FMACH, LINES

DIMENSION NBLUM(1), INSUP(1), NODE(1)
DIMENSION XIQ(1), ETAQ(1), CSG(1)
DIMENSION X3L(1), YL(1), XP(1), YP(1), H(1), DHDX(1)
DIMENSION PHI(MOOF, NMODE, NMODES), PHIZ(NDOFB, 1), PHIZ(NDOFB, 1)
DATA NTP1, NTP2, NTP3, NTP4, NTP5, NTP6, NTP9, 1, 2, 3, 4, 5, 6, 9 /
DATA NTPH, NTPH, NTPH, NTPSH, NTPSH, NTPSH4 / 10, 11, 3, 8 /
DATA CODE / -1 /
10 FORMAT (6I12)
20 FORMAT (1H1, /// 10X, 22H*** SEGMENT 2 *** /// 5X,
1 62HGENERATE -H- AND -DHDX- ARRAYS FOR ALL DEGREES OF
2 FREQUUM
50 FORMAT (1H0 / 6H THE, 14, 26H BODIES OF SUPERBODY NO., 14/)
60 FORMAT (1H0 / 6H THE, 14, 26H PANELS OF SUPERPANEL NO., 14/)
70 FORMAT (1H0 / 2015 )
80 FORMAT (1H0 / 61H SUPER INDEX MODE NODE NODAL POINT
1 COORDINATES / 8H SUPER PANEL, 22X, 3H(1), 6X, 21HXI(1)
90 FORMAT (1H0 / 61H SUPER INDEX MODE NODE NODAL POINT
1 COORDINATES / 8H SUPER BODY, 22X, 3H(1), 6X, 21HXI(1)
120 FORMAT (1H0 / 5X, 14, 33H XP-ELEMENTS FOR SUPERPANEL
130 FORMAT (1H0 / 5X, 14, 33H YP-ELEMENTS FOR SUPERPANEL
140 FORMAT (1H0 / 5X, 14, 33H NO., 14/)
150 FORMAT (1H0 / 5X, 14, 33H NO., 14/)
160 FORMAT (1H0 / 6E16.6 )
IF (IFNEWH.EQ.0) GO TO 180
DO 172 J = 1, NMODES
DO 170 I = 1, NDOFB3
PHIZ(I, J) = 0.0
PHIZ(I, J) = 0.0
170 CONTINUE
172 CONTINUE
180 CONTINUE = 0
K

```

```

MODE      = 0
READ      (NTI,LO) NSB,NSP,NMAX,KPRINT
IF (NMAX.EQ.0) NMAX = MAXO(NTU, NSBETO)
NMAX      = NMAX+3
NLOC      = 1
NEXT      = NMAX * NMAX + 1
NEXT1     = MAXO(NTP,NSBETO+NB) + 1
REWIND    NTP1
REWIND    NTP2
REWIND    NTP3
REWIND    NTP4
REWIND    MASTH
REWIND    MASTH4
REWIND    MASTH4
LMAX      = NTOTAL
NTOT      = 0
KCUM1     = 0
KCUM2     = 0
KCUM3     = 0
LX        = 0
NREOLD    = 0
IF (NB.EQ.0) GO TO 230
DO 220 IB=1,NB
  NBEL     = NSBEA(IB) + 1
  DO 210 IL=1,NBEL
    L      = L + 1
  IF (IL.EQ.NBEL) GO TO 190
  LX      = LX + 1
  X3L(L)  = XIS1(LX)
  GO TO 200
190 CONTINUE
C
  X3L(L)  = XIS2(LX)
200 CONTINUE
210 CONTINUE
  NREOLD  = NREOLD + NBEL
  NRCUM(IB) = NREOLD
220 CONTINUE
230 CONTINUE
SGP      = 1.0
C-P      = 1.0
L        = 0
KS       = 0
L2       = 0
DO 280 IP=1,NP
  NTPH    = NTP4
  NC      = NCARAY(IP)
  NS      = NSARAY(IP)
  DO 270 N=1,NS
    KS     = KS + 1
  LI      = L2 + 1

```

SPLINE50
 SPLINE51
 SPLINE52
 SPLINE53
 SPLINE54
 SPLINE55
 SPLINE56
 SPLINE57
 SPLINE58
 SPLINE59
 SPLINE60
 SPLINE61
 SPLINE62
 SPLINE63
 SPLINE64
 SPLINE65
 SPLINE66
 SPLINE67
 SPLINE68
 SPLINE69
 SPLINE70
 SPLINE71
 SPLINE72
 SPLINE73
 SPLINE74
 SPLINE75
 SPLINE76
 SPLINE77
 SPLINE78
 SPLINE79
 SPLINE80
 SPLINE81
 SPLINE82
 SPLINE83
 SPLINE84
 SPLINE85
 SPLINE86
 SPLINE87
 SPLINE88
 SPLINE89
 SPLINE90
 SPLINE91
 SPLINE92
 SPLINE93
 SPLINE94
 SPLINE95
 SPLINE96
 SPLINE97
 SPLINE98
 SPLINE99
 SPLIN100
 SPLIN101

SPLIN102
SPLIN103
SPLIN104
SPLIN105
SPLIN106
SPLIN107
SPLIN108
SPLIN109
SPLIN110
SPLIN111
SPLIN112
SPLIN113
SPLIN114
SPLIN115
SPLIN116
SPLIN117
SPLIN118
SPLIN119
SPLIN120
SPLIN121
SPLIN122
SPLIN123
SPLIN124
SPLIN125
SPLIN126
SPLIN127
SPLIN128
SPLIN129
SPLIN130
SPLIN131
SPLIN132
SPLIN133
SPLIN134
SPLIN135
SPLIN136
SPLIN137
SPLIN138
SPLIN139
SPLIN140
SPLIN141
SPLIN142
SPLIN143
SPLIN144
SPLIN145
SPLIN146
SPLIN147
SPLIN148
SPLIN149
SPLIN150
SPLIN151
SPLIN152
SPLIN153

```

L2 = L2 + NC
DO 260 L = L1, L2
  CGP = CG(KS)
  IF (ABS(CGP) .LE. 0.00001) GO TO 250
  CSG(IP) = CGP
  YL(L) = YS(KS) / CGP
  GO TO 260
250 SGP = SG(KS)
  CSG(IP) = SGP
  IF (ABS(SGP) .LE. 0.00001) SGP = 1.0

C
YL(L) = ZS(KS) / SGP
260 CONTINUE
270 CONTINUE
280 CONTINUE
REWIND NTPSH
REWIND NTPSH4
WRITE (NTP6,20)
NMODE = 0

C
IMDUNE = 1
IF (NSB .EQ. 0) GO TO 510
NSBP = NSB
NIDT = NBCUM(NB)
300 CONTINUE
ID = 0
DO 490 ISP = 1, NSBP
  IYCON = 0
  IF2 = 0
  READ (NTI,10) IF1, NX3, NSUP, IXCON, NACELL
  IF (IXCON .EQ. 0) IYCON = 1
  READ (NTI,10) (INSUP(I), I=1, NSUP)
  IF (IF1 .NE. 1) GO TO 340
  WRITE (NTP6,50) NSUP, ISP

C
340 CONTINUE
READ (NTI,10) (NODE(I), I=1, NXQ)
IF (IFNEWH .EQ. 0) GO TO 350
DO 342 I = 1, NXQ
  K = K + 1
  L = NODE(I)
DO 346 N = 1, NMODES
  PHIZ(K,N) = PHI(3, L, N)
  PHIZ(K,N) = PHI(3, L, N)
  PHIZ(K,N) = PHI(3, L, N)
346 CONTINUE
IF (NACELL .EQ. 0) GO TO 342
K = K + 1
DO 348 N = 1, NMODES
  PHIZ(K,N) = PHI(5, L, N)
  PHIZ(K,N) = PHI(5, L, N)
  PHIZ(K,N) = PHI(5, L, N)
348 CONTINUE

```

SPLIN154
SPLIN155
SPLIN156
SPLIN157
SPLIN158
SPLIN159
SPLIN160
SPLIN161
SPLIN162
SPLIN163
SPLIN164
SPLIN165
SPLIN166
SPLIN167
SPLIN168
SPLIN169
SPLIN170
SPLIN171
SPLIN172
SPLIN173
SPLIN174
SPLIN175
SPLIN176
SPLIN177
SPLIN178
SPLIN179
SPLIN180
SPLIN181
SPLIN182
SPLIN183
SPLIN184
SPLIN185
SPLIN186
SPLIN187
SPLIN188
SPLIN189
SPLIN190
SPLIN191
SPLIN192
SPLIN193
SPLIN194
SPLIN195
SPLIN196
SPLIN197
SPLIN198
SPLIN199
SPLIN200
SPLIN201
SPLIN202
SPLIN203
SPLIN204
SPLIN205

```

342 CONTINUE
350 CONTINUE
      IF (IF1.EQ.1) GO TO 352
      WRITE (NTP6,60) NSUP,ISP
352  WRITE (NTP6,70) (INSUP(I), I=1,NSUP)
      IF (IF1.EQ.1) GO TO 370
      IPAN = INSUP(1)
      NSRUN = 0
      DO 360 IPN = 1, IPAN
      NSRUN = NSRUN + NSARAY(IPN)
360  CONTINUE
      ABSCOS = ABS(CG(NSRUN))
      WRITE (NTP6,80)
370  CONTINUE
      IF (IF1.EQ.1) WRITE (NTP6,90)
      DO 400 I = 1, NXQ
      ID = ID + 1
      MODE = MODE + 1
      INOD = INOD(I)
      XIQ(I) = ELXI(INOD)
      ETAQ(I) = 0.0
      IF (IF1.EQ.1) GO TO 390
      IF (ABSCOS.LE.0.00001) GO TO 380
      ETAQ(I) = ELYI(INOD)
      GO TO 390
380  CONTINUE
      ETAQ(I) = ELZI(INOD)
390  CONTINUE
      WRITE (NTP6,160) ISP, ID, MODE, NODE(I), XIQ(I), ETAQ(I)
400  CONTINUE
      IF (NACELL.EQ.0) GO TO 410
      NEW NACELLE OPTION -- MAY 1977
      CALL NACL(ISP, NTPS,1, NBEULD, NBCUM, X3L, NXQ, XIQ,
1      NSUP, INSUP, H, JHDX, NMODE, KPRINT, NTP6, )
      GO TO 490
410 CONTINUE
      IF (IMDONE.NE.3) NMODE = NMODE + NXQ
      I1 = 1
      I2 = 1
      ISORT = 1
      CALL
1      SORT(IF1,IF2,NSUP,INSUP,I1,I2,NXQ,LMAX,X,XIC,X3L,YL,
2      XP,YP,H,JHDX,ISP,NSBP,NBCUM,NBARAY,NTPH,NTPDH,
3      NTPSH,KPRINT,NTP6,KCUM1,KCUM2,KCUM3,NTOT,ISORT,NODE,
      WORK(NLOC),WORK(NEXT1))
      IF (KPRINT.NE.2) GO TO 430
      IF (IF1.EQ.1) GO TO 420
      WRITE (NTP6,120) I2,ISP
      WRITE (NTP6,150) (XP(I), I=1,I2)

```

SPLIN206
SPLIN207
SPLIN208
SPLIN209
SPLIN210
SPLIN211
SPLIN212
SPLIN213
SPLIN214
SPLIN215
SPLIN216
SPLIN217
SPLIN218
SPLIN219
SPLIN220
SPLIN221
SPLIN222
SPLIN223
SPLIN224
SPLIN225
SPLIN226
SPLIN227
SPLIN228
SPLIN229
SPLIN230
SPLIN231
SPLIN232
SPLIN233
SPLIN234
SPLIN235
SPLIN236
SPLIN237
SPLIN238
SPLIN239
SPLIN240
SPLIN241
SPLIN242
SPLIN243
SPLIN244
SPLIN245
SPLIN246
SPLIN247
SPLIN248
SPLIN249
SPLIN250
SPLIN251
SPLIN252
SPLIN253
SPLIN254
SPLIN255
SPLIN256
SPLIN257

```

C C C C
      WRITE (NTP6,130) I2, ISP
      WRITE (NTP6,150) (YP(I), I=11,12)
      GO TO 430
420  WRITE (NTP6,140) I2, ISP
      WRITE (NTP6,150) (XP(I), I=11,12)
430  CONTINUE
      REWIND NTP1

C C C C
      ** THE INPUT ARRAY ETAQ IS NOW DIVIDED BY THE 'MODIFIED
      CUSINE' OF THE DIHEDRAL ANGLE OF THE FIRST PANEL IN THE
      CURRENT SUPERPANEL FOR WHICH THE ETA-Q ARRAY IS USED
      IF ((IF1+IF2).GE. 1) GO TO 440
      IQ = INSUP(I)
      COSGAM = CSG(IQ)
      GO TO 450
440  COSGAM = 1.0
450  CONTINUE
      DO 460 IXQ=1,NXQ
      ETAQ(IXQ) = ETAQ(IXQ) / COSGAM
460  CONTINUE
      CALL SPL1( (IF1,NXQ,LMAX,NTP1,NTP6,
      WORK(NLOC), WORK(NEXT)), IXCON, IYCON, NMAX,
      NTPDH,I2,XP,YP,XIQ,ETAQ,H,DHDX,
      WORK(NLOC),WORK(NEXT),NMAX)
      1 CALL SPL2( (IF1,IF2,NXQ,LMAX,NTP2,NTP3,NTPH,
      NTPDH,I2,XP,YP,XIQ,ETAQ,H,DHDX,
      WORK(NLOC),WORK(NEXT),NMAX)
      2 ISORT = 2
      REWIND NTPH
      REWIND NTPDH
      NTPSH = NTP9
      IF (IMDUNE .EQ. 3) NTPSH = NTPSH4
      CALL SORT( (IF1,IF2,NSUP,INSUP,I1,I2,NXQ,LMAX,X,
      XIC,X3L,YL,
      XP,YP,H,DHDX,ISP,NSBP,NBCUM,NBARAY,NTPH,NTPDH,
      NTPSH,KPRINT,NTP6,KCUM1,KCUM2,KCUM3,NTOT,ISORT,NODE,
      WORK(NLOC),WORK(NEXT))
      1 NTPSH = NTP9
      2 GO TO (490,470,480), IMDUNE
      3 NTPSH = NTP9
      GO TO (490,470,480), IMDUNE
470  CONTINUE
      IMDUNE = 3
      IF2 = 1
      GO TO 410
480  CONTINUE
      IMDUNE = 2
490  CONTINUE
      IF (IMDUNE .GT. 1) GO TO 520

C C C C
      ** COPY CONTENTS OF TAPE NTPSH (H AND DHDX MATRIX COLUMNS
      FOR ALL BODY ELEMENTS AND ALL 'MODES') ONTO THE MASTER
      TAPES MASTH AND MASTDH RESPECTIVELY, EACH PRECEDED BY THE
      TOTAL NUMBER OF MODES, NMODE.
      REWIND NTPSH
      NMODE = NMODE

```

SPLIN258
SPLIN259
SPLIN260
SPLIN261
SPLIN262
SPLIN263
SPLIN264
SPLIN265
SPLIN266
SPLIN267
SPLIN268
SPLIN269
SPLIN270
SPLIN271
SPLIN272
SPLIN273
SPLIN274
SPLIN275
SPLIN276
SPLIN277
SPLIN278
SPLIN279
SPLIN280
SPLIN281
SPLIN282
SPLIN283
SPLIN284
SPLIN285
SPLIN286
SPLIN287
SPLIN288
SPLIN289
SPLIN290
SPLIN291
SPLIN292
SPLIN293
SPLIN294
SPLIN295
SPLIN296
SPLIN297
SPLIN298
SPLIN299
SPLIN300
SPLIN301
SPLIN302
SPLIN303
SPLIN304
SPLIN305
SPLIN306
SPLIN307

```

DO 500 II = 1, NMODE
  READ (NTPSH) NODEK, NTOT, ( H (I), I = 1, NTOT)
  IF (IFNEWH.NE.0) NODEK = II
  WRITE (MASTH) NODEK
  READ (NTPSH) NTOT, ( H (I), I = 1, NTOT)
  WRITE (MASTH) NTOT, (DHDX(I), I = 1, NTOT)
  READ (NTPSH) NTOT, (DHDX(I), I = 1, NTOT)
  WRITE (MASTH) NTOT, (DHDX(I), I = 1, NTOT)
500 CONTINUE
  WRITE (MASTH) KODE
  WRITE (MASTH) KODE
  REWIND NTPSH
  REWIND NTPSH4
  NMODE = 0
510 CONTINUE
  IF (NSP.EQ.0.OR. IFNEWH.NE.0) GO TO 540
  NSBP = NSP
  NTOT = NTP
  IMDONE = 2
  GO TO 300
520 CONTINUE
  *** COPY CONTENTS OF TAPE NTPSH (H AND DHDX MATRIX COLUMNS
  FOR ALL LIFTING SURFACE BOXES AND ALL MODES) ONTO THE
  MASTER TAPES MASTH AND MASTH4 RESPECTIVELY PRECEDED
  BY THE NUMBER OF MODES NMODE -- ALSO
  COPY H-1/4 COLUMNS FROM NTPSH4 ONTO THE MASTER TAPE
  MASTH4, PRECEDED BY NMODE.
  REWIND NTPSH
  REWIND NTPSH4
  DO 530 II = 1, NMODE
    READ (NTPSH) NODEK, NTOT, ( H (I), I = 1, NTOT)
    WRITE (MASTH) NODEK
    READ (MASTH) NTOT, ( H (I), I = 1, NTOT)
    WRITE (NTPSH) NTOT, (DHDX(I), I = 1, NTOT)
    READ (MASTH) NTOT, (DHDX(I), I = 1, NTOT)
    WRITE (NTPSH4) NTOT, ( H (I), I = 1, NTOT)
    WRITE (MASTH4) NTOT, ( H (I), I = 1, NTOT)
530 CONTINUE
    WRITE (MASTH) KODE
    WRITE (MASTH) KODE
    WRITE (MASTH4) KODE
540 CONTINUE
  RETURN
  END

```

```

1 SUBROUTINE SPLN(SPLN, SCOFF, THB, NODES, XI, IPRINT, NTP6)
2   XA, YA, XB, YB, NDIM, NTP6
3   )
4   THIS SUBROUTINE COMPUTES THE *SPLINE* MATRIX, AND THE *RHS*
5   MATRIX FOR A SUPERPANEL, THEN CALL SUBROUTINE MIS1 TO SOLVE
6   FOR THE SPLINE COEFFICIENTS MATRIX *SCOFF*
7
8   DIMENSION SPLN(NDIM, NDIM), SCOFF(NDIM, NODES), THB(1),
9   XI(1), ETA(1)
10
11   NODP1 = NODES + 1
12   NODP2 = NODES + 2
13   NODM1 = NODES - 1
14
15   IF (NODES .EQ. 1) GO TO 140
16   DO 20 J = 1, 2
17     DO 10 I = 1, 2
18       SPLINE(I, J) = 0.0
19     CONTINUE
20 CONTINUE
21
22   DO 60 I = 1, NODP2
23     IP2 = I + 2
24     SPLINE(IP2, 1) = 1.0
25     SPLINE(1, IP2) = 1.0
26     IF (I .GT. 1 .AND. I .LT. NODP2) GO TO 40
27     IF (I .EQ. NODP2) GO TO 30
28     THB(I) = 0.0
29     GO TO 50
30 CONTINUE
31 THB(1) = SQRT((XB-XA)**2 + (YB-YA)**2)
32 GO TO 50
33 CONTINUE
34 J = 1
35 THB(J) = 1 - 1
36 CONTINUE
37 THB(J) = SQRT((XI(J)-XA)**2 + (ETA(J)-YA)**2)
38 CONTINUE
39 SPLINE(IP2, 2) = THB(1)
40 SPLINE(2, IP2) = THB(1)
41 CONTINUE
42 DO 80 J = 1, NODP1
43   JP1 = J + 1
44   JP2 = J + 2
45   K = JP1
46   DO 70 I = K, NODP2
47     IP2 = I + 2
48     ARG = ABS(THB(I) - THB(J))
49     AIJ = 0.0
50     IF (ARG .LT. .0001) GO TO 62
51     ARG2 = ARG**2
52
53

```

```

        AIJ = 2.0*ARG2 * ALOG(ARG)
62  CONTINUE
    SPLINE(IP2,JP2) = AIJ
70  CONTINUE
80  CONTINUE
C
C      COMPUTE MATRIX OF RIGHT-HAND-SIDES (SAVE IN ARRAY *SCOE*)
C
    B = -THB(2) / (THB(3)-THB(2))
    A = 1.0 - B
    D = (THB(NODP2)-THB(NODES)) / (THB(NDDP1)-THB(NODES))
    C = 1.0 - D
C
    SCOE(3,1) = A
    SCOE(3,2) = B
    SCOE(NDIM,NODM1) = C
    SCOE(NDIM,NODES) = D
C
    DO 90 I = 1, NODES
    IP3 = I + 3
    SCOE(IP3,1) = 1.0
90  CONTINUE
C
C      IF (IPRINT .NE. 2) GO TO 130
    LINES = 45
    LIMIT = LINES
    LCUM = 3
    WRITE (NTP6,110)
    DO 100 J = 1, NDIM
    WRITE (NTP6,120) J, (SPLINE(I,J), I = 1, NDIM)
    LCUM = LCUM + 5 + NDIM / 6
    IF (LCUM .LT. LIMIT .OR. J .EQ. NDIM) GO TO 100
    WRITE (NTP6,110)
    LCUM = LCUM + 3
    LIMIT = LIMIT + LINES
100 CONTINUE
C
110 FORMAT (1H1,/,/,10X,25H*** SPLINE MATRIX *** /
120 FORMAT (1/10X,6HCOLUMN,13// (5X,6E15.6)
C
130 CONTINUE
C
    SOLVE FOR THE SPLINE COEFFICIENTS *SCOE*
C
    NMAX = NDIM
C
    CALL MISL(SPLINE, NDIM, NMAX, SCOE, NODES, NERR, 0)
C
    IF (IPRINT .EQ. 2) GO TO 160

```

SPLN 54
 SPLN 55
 SPLN 56
 SPLN 57
 SPLN 58
 SPLN 59
 SPLN 60
 SPLN 61
 SPLN 62
 SPLN 63
 SPLN 64
 SPLN 65
 SPLN 66
 SPLN 67
 SPLN 68
 SPLN 69
 SPLN 70
 SPLN 71
 SPLN 72
 SPLN 73
 SPLN 74
 SPLN 75
 SPLN 76
 SPLN 77
 SPLN 78
 SPLN 79
 SPLN 80
 SPLN 81
 SPLN 82
 SPLN 83
 SPLN 84
 SPLN 85
 SPLN 86
 SPLN 87
 SPLN 88
 SPLN 89
 SPLN 90
 SPLN 91
 SPLN 92
 SPLN 93
 SPLN 94
 SPLN 95
 SPLN 96
 SPLN 97
 SPLN 98
 SPLN 99
 SPLN 100
 SPLN 101
 SPLN 102
 SPLN 103
 SPLN 104
 SPLN 105

```

C 140 CONTINUE
      SCOE(1,1) = 1.0
      DO 150 I = 2, 5
C 150 SCOE(I,1) = 0.0
C
C 160 CONTINUE
      RETURN
C
      LINES = 45
      LIMIT = LINES
      LCUM = 3
      WRITE (NTP6,180)
      DO 170 J = 1, NODES
      WRITE (NTP6,190) J, (SCOE(I,J), I = 1, NDIM)
      LCUM = LCUM + 5 + NDIM / 6
      IF (LCUM .LT. LIMIT .OR. J .EQ. NODES) GO TO 170
      WRITE (NTP6,180)
      LIMIT = LIMIT + LINES
      LCUM = LCUM + 3
      CONTINUE
C 170
C 180 FORMAT ( 1H1 /// 10X, 37H*** SPLINE COEFFICIENT MATRIX *** )
C 190 FORMAT ( / 10X, 6HCOLUMN, 13 // ( 5X, 6E15.6 )
      RETURN
      END

```

```

SPLN 106
SPLN 107
SPLN 108
SPLN 109
SPLN 110
SPLN 111
SPLN 112
SPLN 113
SPLN 114
SPLN 115
SPLN 116
SPLN 117
SPLN 118
SPLN 119
SPLN 120
SPLN 121
SPLN 122
SPLN 123
SPLN 124
SPLN 125
SPLN 126
SPLN 127
SPLN 128
SPLN 129
SPLN 130
SPLN 131
SPLN 132
SPLN 133

```

```

SUBROUTINE SPLI(IFI,NXQ,LMAX,NTPI,NTP6,
*           RHS,XKD,IXCON,IYCON,NMAX
*           DIMENSION XKD(NMAX,NMAX); RHS(NMAX,NMAX)
*           DIMENSION XIQ(1); ETAQ(1)
*           IFLAG = 0
*           NUMBR = 0
*           NKD = NXQ + 3
*           BFLAG = 0.0
*           COSAV = 0.0
*           SINAV = 0.0
*           C2TIP1 = 0.0
*           S2TIP1 = 0.0
*           DD 70 IQ = 2, NXQ
*           DELX2 = ( XIQ(IQ) - XIQ(1) ) ** 2
*           DELY2 = ( ETAQ(IQ) - ETAQ(1) ) ** 2
*           RHO = DELX2 + DELY2
*           C2TI = DELX2 / RHO
*           S2TI = DELY2 / RHO
*           IF ( IQ .EQ. 2 ) GO TO 60
*           EPS1 = ABS(C2TIP1 - C2TI)
*           EPS2 = ABS(S2TIP1 - S2TI)
*           IF ( EPS1 .GE. 0.01 .OR. EPS2 .GE. 0.01 ) GO TO 80
*           C2TIP1 = C2TI
*           S2TIP1 = S2TI
*           COSAV = COSAV + C2TI
*           SINAV = SINAV + S2TI
*           CONTINUE
*           SIGNX = 1.0
*           SIGNY = 1.0
*           XIDIF = XIQ(2) - XIQ(1)
*           ETDIF = ETAQ(2) - ETAQ(1)
*           IF (XIDIF .EQ. 0.0) GO TO 72
*           IF (XIDIF .EQ. 0.0) GO TO 75
*           SIGNX = XIDIF / ABS(XIDIF)
*           SIGNY = ETDIF / ABS(ETDIF)
*           CONTINUE
*           CTH = SIGNX * SQR(COSAV / (NXQ - 1))
*           STH = SIGNY * SQR(SINAV / (NXQ - 1))
*           BFLAG = 1.0
*           CONTINUE
*           J=1,3
*           DO 200 XKD(1,J) = 0.0
*           IF (IFI .EQ. 0) GO TO 120
*           IF (J .EQ. 1) OR (J .EQ. 3) GO TO 90
*           IF (J .EQ. 2) GO TO 90
*           XKD(2,J) = -STH
*           GO TO 150
*           CONTINUE
*           XKD(2,J) = -CTH
*           XKD(3,J) = -CTH

```

C

SPLI 54
SPLI 55
SPLI 56
SPLI 57
SPLI 58
SPLI 59
SPLI 60
SPLI 61
SPLI 62
SPLI 63
SPLI 64
SPLI 65
SPLI 66
SPLI 67
SPLI 68
SPLI 69
SPLI 70
SPLI 71
SPLI 72
SPLI 73
SPLI 74
SPLI 75
SPLI 76
SPLI 77
SPLI 78
SPLI 79
SPLI 80
SPLI 81
SPLI 82
SPLI 83
SPLI 84
SPLI 85
SPLI 86
SPLI 87
SPLI 88
SPLI 89
SPLI 90
SPLI 91
SPLI 92
SPLI 93
SPLI 94
SPLI 95
SPLI 96
SPLI 97
SPLI 98
SPLI 99
SPLI 100
SPLI 101
SPLI 102
SPLI 103
SPLI 104
SPLI 105

```

105 GO TO 150
110 XKD(2,J) = 0.0
110 XKD(3,J) = 0.0
120 IF (IXCON .EQ. I : AND : J : EQ .2) GO TO 130
120 IF (IYCON .EQ. I : AND : J : EQ .3) GO TO 140
130 XKD(2,J) = 1.0
140 XKD(2,J) = 0.0
140 XKD(3,J) = 1.0
150 CONTINUE
DO 190 I=4,NKD
IM3 = I-3
GO TO (160, 170, 180), J
160 XKD(I,J) = 1.0
170 GO TO 190
170 XKD(I,J) = XIQ( IM3)
180 XKD(I,J) = ETAQ(IM3)
190 CONTINUE
C *** WRITE (NTPI)
      ONE COLUMN OF THE KD- MATRIX (OUT OF THE FIRST 3)
      (XKD(I,J), I=1,NKD)
200 CONTINUE
DO 300 J=4,NKD
JM3 = J-3
XKD(1,J) = 1.0
XKD(2,J) = XIQ( JM3)
XKD(3,J) = ETAQ(JM3)
DO 260 I=4,NKD
IM3 = I-3
IF (I .EQ. J) GO TO 250
IF (XIQ(IM3) .EQ. XIQ(JM3)) GO TO 220
TRM1 = (XIQ(IM3)-XIQ(JM3))*2
210 IF (ETAQ(IM3) .EQ. ETAQ(JM3)) GO TO 230
TRM2 = (ETAQ(IM3)-ETAQ(JM3))*2
GO TO 240
220 TRM1 = 0.0
230 TRM2 = 0.0
240 RAIJ2 = TRM1 + TRM2
AIJ = RAIJ2 * ALOG(RAIJ2)
GU TO 260
250 AIJ = 0.0
260 XKD(I,J) = AIJ
C *** WRITE (NTPI)
      A COLUMN OF THE KD-MATRIX
      (XKD(I,J), I=1,NKD)
300 CONTINUE
C *** KD MATRIX FORMATION FOR ONE SUPERPANEL (BODY) COMPLETE
C *** FORM RHS OF EQ. XXX

```

106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125

```

C      DO 700 J=1,NXQ
      DO 650 I=1,NKD
      IM3 = I - 3
      RHS(I,J) = 0.0
      IF (IM3 .EQ. J) RHS(I,J) = 1.0
      650 CONTINUE
      660 CONTINUE
      700 CONTINUE
C      *** USE THE STANDARD SUBROUTINE MIS1 TO SOLVE EQ. XXX
C      NERR = 0
C      SCALR = 0.0
C      NKMAX = NMAX
C      CALL MIS1(NKD, NKD, NKMAX, RHS, NXQ, NERR, SCALR)
C
C      RETURN
C      END
  
```

```

SUBROUTINE SPL2(IF1,IF2,NXQ,LMAX,NTP2,NTP3,NTP6,KPRINT,NTPH,
NTPDH,I2,XP,YP,XIQ,ETAQ,XKF,DKFDX,CMAT,SUM,NMAX)
1 DIMENSION XP(LMAX),YP(LMAX),XIQL(1),ETAQL(1),
DIMENSION XKF(1),DKFDX(1)
DIMENSION SUM(I2,NMAX),CMAT(NMAX,NMAX)
25 FORMAT (6E16.6)
35 FORMAT (1H1 /// 6H TFE, I4, 3H BY, I4, 24H H (3/4-CHORD)
MATSP2
78 IRIX / )
90 IRIX / )
100 IRIX / )
110 IRIX / )
120 IRIX / )
130 IRIX / )
140 IRIX / )
150 IRIX / )
160 IRIX / )
170 IRIX / )
180 IRIX / )
190 IRIX / )
200 IRIX / )
210 IRIX / )
220 IRIX / )
230 IRIX / )
240 IRIX / )
250 IRIX / )
260 IRIX / )
270 IRIX / )
280 IRIX / )
290 IRIX / )
300 IRIX / )
310 IRIX / )
320 IRIX / )
330 IRIX / )
340 IRIX / )
350 IRIX / )
360 IRIX / )
370 IRIX / )
380 IRIX / )
390 IRIX / )
400 IRIX / )
410 IRIX / )
420 IRIX / )
430 IRIX / )
440 IRIX / )
450 IRIX / )
460 IRIX / )
470 IRIX / )
480 IRIX / )
490 IRIX / )
500 IRIX / )
510 IRIX / )
520 IRIX / )
530 IRIX / )

```


SPL2 106
SPL2 107
SPL2 108
SPL2 109
SPL2 110
SPL2 111
SPL2 112
SPL2 113
SPL2 114
SPL2 115
SPL2 116
SPL2 117
SPL2 118
SPL2 119
SPL2 120
SPL2 121
SPL2 122
SPL2 123
SPL2 124
SPL2 125
SPL2 126
SPL2 127
SPL2 128
SPL2 129
SPL2 130
SPL2 131
SPL2 132
SPL2 133
SPL2 134

```

REWIND NTP3
DO 720 J=1,NXQ
DO 710 I=1,NN
710 SUM(I,J) = 0.0
720 CONTINUE
DO 750 J=1,NKF
READ (NTP3) (DKFDX(II), II=1,NN)
DO 740 I=1,NN
DO 730 JC=1,NXQ
SUM(I,JC) = SUM(I,JC) + DKFDX(I) * CMAT(J,JC)
730 CONTINUE
740 CONTINUE
750 CONTINUE
C *** COMPUTATION OF THE -DHDX- MATRIX COMPLETE
C *** WRITE MATRIX COLUMNS ON TAPE NTPUH
C
IF (KPRINT.NE.0) WRITE (NTP6,55) NN, NXQ
DO 770 J=1,NXQ
WRITE (NTPDH) (SUM(I,J), I=1,NN)
IF (KPRINT.EQ.0) GO TO 770
WRITE (NTP6,60) J
WRITE (NTP6,25) (SUM(I,J), I=1,NN)
770 CONTINUE
C 800 CONTINUE
RETURN
END

```

```

SUBROUTINE SUBB(KB,KS,I,J,JZ,JB,LB,LS,NDY,NYFL,FLND,FLNE,PI,EPS,
1 SGR,CGR,SGS,AK,SL,CL,TL,FL,BETA,SUM
2 YB,ZR,RIA,X,Y,ZS,DELX)
3 DIMENSION YB(1),ZB(1),RIA(1),X(1),YS(1),ZS(1),DELX(1)
4 ** COMPUTES ELEMENTS OF THE SUBMATRICES DZP, DZZ, DZY, DYP,
5 DYZ AND DYY USING SUBROUTINE DZY
6 COMMON /AROCOM/ NTI, MUDS
7 X, NP, MSTRIP, NSMAX, NCMAX, NTOIAL, NB, MSBE, MBE
8 Y, ND, NE, NBY, NBZ, NTO, NIP, NTV, NTZ
9 I, NTYS, NTZS, MAXGR, MAXSTR, NSBEIO, NSTRIP, KR, XM, REFA, REFC
10 KEFS, FMACH, LINES
11 COMMON /KCS/ IND, KD1P, KD1I, KD2R, KD2I
12 REAL KDIR, KD1I, KD2R, KD2I
13 REAL KR, M
14 COMPLEX DPUR, DPUL, DPLR, DPLL, DP, SUM(2), EIKX
15 FORMAT (1H0, 6E20.8)
16 IPRINT = 0
17 IND = 0
18 BR = FL/2.0
19 DPUR = (0.0,0.0)
20 DPUL = (0.0,0.0)
21 DPLR = (0.0,0.0)
22 DPLL = (0.0,0.0)
23 ANDT = RTA(JB)
24 DXS = DELX(J)
25 ABSYB= ABS(YB(LB))
26 ABSZR= ABS(ZB(LB))
27 IF (KB.EQ.0) GO TO 20
28 TEST1= ABS(YB(LB) - YB(KB))
29 TEST2= ABS(ZB(LB) - ZB(KB))
30 IF (TEST1.GT.EPS. OR .TEST2.GT.EPS) GO TO 20
31 SUM(1) = (0.0,0.0)
32 SUM(2) = (0.0,0.0)
33 IF (NDY.NE.NYFL) GO TO 100
34 IF (I.NE.J) GO TO 100
35 D2D = 1.0 / (2.0*PI*ANOT*ANOT*(1.0+AR))
36 IF (NDY.NE.0) D2D=D2D/AR
37 SUM(1)= CMPLX(D2D,0.0)
38 SUM(2)= CMPLX(D2D,0.0)
39 GO TO 100
40 CONTINUE
41 XX = X(I)
42 YY = YS(KS)
43 Z = ZS(KS)
44 XI1 = X(J) - 0.5*DXS
45 XI2 = X(J) + 0.5*DXS
46 ETA = YS(LS)
47 ZETA = ZS(LS)
48 AD = ANOT
49 IDZDY = NDY

```

```

      IGO = 1 = 0
      LHS CONTINUE
30  CALL      DZY (XX, Y, Z, SGR, CGR, XII, XI2, ETA, ZETA, AR, AU,
1      KR      IDZDY, REF, BETA, FMACH, LHS, I^RNT )
2      DP = CMPLX(DZDYR, DZDYI)
      GO TO (40, 50, 70, 80), IGO
C 40 CONTINUE
      UPPER RIGHT-HAND SIDE CONTRIBUTION
      DPUR = DP
      IF (KB.EQ.LB) GO TO 100
      IF (ABSXB.LE.EPS.AND.ABSZB.LE.EPS) GO TO 90
      IF (ABSXB.LE.EPS) GO TO 60
      IGO = 2
      ETA = -YS(LS)
      LHS = 1
      GO TO 30
C 50 CONTINUE
      UPPER LEFT-HAND SIDE CONTRIBUTION
      DPUL = DP
      LHS = 0
      GO TO 30
C 60 CONTINUE
      IF (NE.EQ.O.OR.ABSZB.LE.EPS) GO TO 90
      IGO = 3
      ETA = YS(LS)
      ZETA = -ZS(LS)
      GO TO 30
C 70 CONTINUE
      LOWER RIGHT-HAND SIDE CONTRIBUTION
      DPLR = DP
      IF (ABSXB.LE.EPS) GO TO 90
      IGO = 4
      ETA = -YS(LS)
      ZETA = -ZS(LS)
      LHS = 1
      GO TO 30
C 80 CONTINUE
      LOWER LEFT-HAND SIDE CONTRIBUTION
      DPLL = DP
      GO TO 30
C 90 CONTINUE
      SUM(1)=DPUR + DPUL + FLNE*DPLK + FLNE*DPLL
      SUM(2)=DPUR - DPUL + FLNE*DPLR - FLNE*DPLL
100 CONTINUE
      RETURN
      END

```

SUBB 50
 SUBB 51
 SUBB 52
 SUBB 53
 SUBB 54
 SUBB 55
 SUBB 56
 SUBB 57
 SUBB 58
 SUBB 59
 SUBB 60
 SUBB 61
 SUBB 62
 SUBB 63
 SUBB 64
 SUBB 65
 SUBB 66
 SUBB 67
 SUBB 68
 SUBB 69
 SUBB 70
 SUBB 71
 SUBB 72
 SUBB 73
 SUBB 74
 SUBB 75
 SUBB 76
 SUBB 77
 SUBB 78
 SUBB 79
 SUBB 80
 SUBB 81
 SUBB 82
 SUBB 83
 SUBB 84
 SUBB 85
 SUBB 86
 SUBB 87
 SUBB 88
 SUBB 89
 SUBB 90
 SUBB 91
 SUBB 92
 SUBB 93
 SUBB 94
 SUBB 95

```

SUBROUTINE SUBI(DA,DZB,DYB,DAR,DETA,DZETA,DCGAM,DEE,DXI,TL,
1 DETAI,DZETAI,DCGAMI,DEEI,DTLAMI,DMUY,DMUZ,INFL,IOUTFL)
2 *** COMPUTES THE IMAGE POINT COORDINATES INSIDE ASSOCIATED
3 BODIES AND THE MU-Z MU-Y ELEMENTS USED IN SUBROUTINE FMMW
4
5 15 FORMAT (1H0,15I5)
6 EPS = 0.1*DEE
7 DMUY = 0.0
8 DMUZ = 0.0
9 IGO = 1
10 PSQR = SQRT(((DETA-DYB)*DAR)**2 + (DZETA-DZB)**2)
11 COSTH = (DETA-DYB)*DAR/PSQR
12 SINTH = (DZETA-DZB)/PSQR
13 CT2 = COSTH*COSTH
14 ST2 = SINTH*SINTH
15 CT3 = COSTH*CT2
16 ST3 = SINTH*ST2
17 YCBAR = DA*(1.0-DAR*DAR)*CT3 + DYB
18 ZCBAR = DA*(DAR*DAR-1.0)*ST3/DAR + DZB
19 PAREN = ST2 + DAR*DAR*CT2
20 PAR3 = PAREN*PAREN**2
21 ABAR = DA*SQRT(PAR3)/DAR
22 ABAR2 = ABAR*ABAR
23 IF (INFL.NE.0) GO TO 300
24 ETA1 = DETA - DEE*DCGAM
25 ETA2 = DETA + DEE*DCGAM
26 ZETA1 = DZETA - DEE*DCGAM
27 ZETA2 = DZETA + DEE*DCGAM
28 RHO22 = (ETA1 - YCBAR)**2 + (ZETA1-ZCBAR)**2
29 RHO22 = (ETA2 - YCBAR)**2 + (ZETA2-ZCBAR)**2
30 ETA11 = YCBAR + (ETA1-YCBAR)*ABAR2/RHO12
31 ETA12 = YCBAR + (ETA2-YCBAR)*ABAR2/RHO12
32 ZET11 = ZCBAR + (ZETA1-ZCBAR)*ABAR2/RHO12
33 ZET12 = ZCBAR + (ZETA2-ZCBAR)*ABAR2/RHO12
34 DEEI = SQRT((ETA12-ETA11)**2 + (ZET12-ZET11)**2) / 2.0
35 DETAI = (ETA11 + ETA12)/2.0
36 DZETAI = (ZET11 + ZET12)/2.0
37 DCGAMI = -(ZET12 - ZET11)/(2.0*DEEI)
38 DXI1 = DXI - DEE*TL
39 DXI2 = DXI + DEE*TL
40 DTLAMI = DELXI/(2.0*DEEI)
41 IF ((ABS(DAR-1.0)).LE.0.0001) GO TO 420
42 GO TO 301
43 300 CONTINUE
44 RHO2 = (DETA - YCBAR)**2 + (DZETA - ZCBAR)**2
45 RHO4 = RHO2*RHO2
46 DETAI = YCBAR + (DETA - YCBAR)*ABAR2/RHO2
47 DZETAI = ZCBAR + (DZETA - ZCBAR)*ABAR2/RHO2
48 CONTINUE
49 GO TO (302,303,304), IGO
50
51
52
53

```



```

C   UPPER RIGHT SENDING POINT
    IGO = 1
    LHS = 0
    NDRLF = 0
    TL = XLAM(J)
    SQT1 = SQR(1.0+TL**2)
    SL = TL / SQT1
    CL = 1.0 / SQT1
    XO = X(I) - XIC(J)
    YO = YREC - YS(LS)
    ZO = ZREC - ZS(LS)
    ES = EE(LS)
    DXS = DELX(J)
    AX = XO
    AY = YO
    AZ = ZO
    CV = DXS
30  CONTINUE
    IMG = 0
    NDR1 = 1
    NA2 = 0
    CALL SNPDF(SL,CL,TL,SGS,CGS,SGR,CGP,XO,YO,ZO,ES,DIJ,BETA,CV)
    IF (KR.LE.EPS) GO TO 40
    SDELX = 2.0*ES
    DELY = AX + ES*TL
    AX1 = AX + ES*CGS
    AY1 = AY + ES*SGS
    AZ1 = AX - ES*TL
    AX2 = AX - ES*TL
    AY2 = AY - ES*CGS
    AZ2 = AZ - ES*SGS
    CALL INCR01(AX,AY,AZ,AX1,AY1,AZ1,AX2,AY2,AZ2,SGR,CGR,SGS,CGS,
1    KK,FL,BETA,SDELX,DELY,DELR,DELI,IO,IR,NBXS,NCPNB,LHS,
2    NDRLE,IMG,NDR1,IMGS,WORK(NWK1),WORK(NWK2),WORK(NWK3),
3    WORK(NWK4),WORK(NWK5),WORK(NWK6),WORK(NWK7),WORK(NWK8))
40  CONTINUE
    IF (NR.EQ.0) GO TO 12C
    ** CHECK FOR ASSOCIATED BODIES
    DIJS = DIJ
    DELRS = DELR
    DELIS = DELI
    DIJI = 0.0
    DELRI = 0.0
    DELII = 0.0
    NA1 = 1
    NA2 = NB
    IMG = 1
    ** START DO-LOOP FOR THE SUMMATION OF THE WING-IMAGE CONTRIBUTIONS
C   OVER RANGE(P)
    DO 110 NA=NA1,NA2
    NUB = NA
C   ** NOB IS THE SEQUENCE NUMBER OF THE CURRENT BODY ASSOCIATED WITH

```

SUBP 50
SUBP 51
SUBP 52
SUBP 53
SUBP 54
SUBP 55
SUBP 56
SUBP 57
SUBP 58
SUBP 59
SUBP 60
SUBP 61
SUBP 62
SUBP 63
SUBP 64
SUBP 65
SUBP 66
SUBP 67
SUBP 68
SUBP 69
SUBP 70
SUBP 71
SUBP 72
SUBP 73
SUBP 74
SUBP 75
SUBP 76
SUBP 77
SUBP 78
SUBP 79
SUBP 80
SUBP 81
SUBP 82
SUBP 83
SUBP 84
SUBP 85
SUBP 86
SUBP 87
SUBP 88
SUBP 89
SUBP 90
SUBP 91
SUBP 92
SUBP 93
SUBP 94
SUBP 95
SUBP 96
SUBP 97
SUBP 98
SUBP 99
SUBP 100
SUBP 101

```

C  PANEL  L  IN WHICH THE SENDING POINT  J  LIES
NOBI = NOB
DA = AVR(NOB)
DAR = ARH(NOB)
DXLE = XLE(NOB)
DXTE = XTE(NOB)
GO TO (50,60,70,80), 160
50 CONTINUE
DZB = ZB(NOB)
DYB = YB(NOB)
DETA = YS(LS)
DZETA = ZS(LS)
GO TO 90
60 CONTINUE
DZB = ZB(NOB)
DYB = YB(NOB)
DETA = YS(LS)
DZETA = ZS(LS)
GO TO 90
70 CONTINUE
DZB = ZB(NOB)
DYB = YB(NOB)
DETA = YS(LS)
DZETA = ZS(LS)
GO TO 90
80 CONTINUE
DZB = ZB(NOB)
DYB = YB(NOB)
DETA = YS(LS)
DZETA = ZS(LS)
GO TO 90
90 CONTINUE
DCGAM = CGS
DSGAM = SGS
DEE = ES
DXI = XIC(J)
IF (DXI.LT.DXLE.OR.DXI.GT.DXTE) GO TO 110
TEST = SQRT((DETA-DYB)**2 + (DZETA-DZB)**2)
IF (TEST.GT.SCALER * DA) GO TO 110
CALL DETA1,DZETA1,DCGAMI,DSGAMI,DEE1,DTLAMI,DMUY,DMUZ,INFL,IOUTFL)
1 DETA1,DZETA1,DCGAMI,DSGAMI,DEE,DXI,TL,
DIJ = 0.0
IF (INFL.NE.0.OR.IOUTFL.EQ.0) GO TO 100
DTL = DTLAMI
DSQRTL = SQRT(1.0+DTL**2)
DSL = DTL/DSQRTL
DCL = 1.0/DSQRTL
XOI = X0
YOI = YREC - DETA1
ZOI = ZREC - DZETA1
CALL SNPDF(DSL,DCL,DTL,DSGAMI,DCGAMI,SCR,CGR,XOI,YOI,ZOI,
1 DFI,DIJ,BETA,CV)
1 DIJ = DIJ+DIJ

```

SUBP 102
SUBP 103
SUBP 104
SUBP 105
SUBP 106
SUBP 107
SUBP 108
SUBP 109
SUBP 110
SUBP 111
SUBP 112
SUBP 113
SUBP 114
SUBP 115
SUBP 116
SUBP 117
SUBP 118
SUBP 119
SUBP 120
SUBP 121
SUBP 122
SUBP 123
SUBP 124
SUBP 125
SUBP 126
SUBP 127
SUBP 128
SUBP 129
SUBP 130
SUBP 131
SUBP 132
SUBP 133
SUBP 134
SUBP 135
SUBP 136
SUBP 137
SUBP 138
SUBP 139
SUBP 140
SUBP 141
SUBP 142
SUBP 143
SUBP 144
SUBP 145
SUBP 146
SUBP 147
SUBP 148
SUBP 149
SUBP 150
SUBP 151
SUBP 152
SUBP 153

```

IF (KR.LE.EPS) GO TO 100
DELR = 0.0
DELI = 0.0
AYI = YOI
AZI = ZOI
AYII = AYI - DEEI*DCGAMI
AZII = AZI - DEEI*DCGAMI
AY2I = AYI + DEEI*DCGAMI
AZ2I = AZI + DEEI*DCGAMI
DEEI2 = 2.0*DEEI
CALL INCRO(AX,AYI,AZI,AXI,AYII,AZII,AX2,AY2I,AZ2I,SGR,CGR,
1 USGAMI,DCGAMI,KR,FL,BETA,SDELX,DEEI2,DELR,DELI,IO,IR,NBXS,
2 NCPNB,LHS,NDBLE,IMG,NDBI,IMGS,WORK(NWK1),WORK(NWK2),WORK(NWK3),
3 WORK(NWK4),WORK(NWK5),WORK(NWK6),WORK(NWK7),WORK(NWK8))
DELR = DELRI+DELI
DELI = DELII+DELI
IF (IO.EQ.NCARAY(L)) IMGS=1
GO TO 110
100 CONTINUE
DELR = 0.0
DELI = 0.0
110 CONTINUE
DIJ = DIJS
DELR = DFLRS
DELI = DELIS
120 CONTINUE
DP = CMPLX((DIJ+DIJI)-(DELR+DELR),(-DELI-DELI))
130 CONTINUE
GO TO (140,150,170,180), IGO
140 CONTINUE
DPUR = DP
IF (ABS(YS(LS))) .LE.0.001) GO TO 160
C UPPER LEFT SENDING POINT
IGO = 2
LHS = 1
SGS = -SGS
TL = -TL
SL = -SL
YO = YRFC + YS(LS)
AY = YO
GO TO 30
150 CONTINUE
DPUL = DP
160 CONTINUE
IF (NE.EQ.0.OR.(ABS(ZS(LS))) .LE.0.001) GO TO 190
C LOWER RIGHT SENDING POINT
IGO = 3
LHS = 0
NDBLE = 1
TL = XLAM(J)
SL = TL/(SCRT(1.0+TL*TL))
YO = YRFC -YS(LS)

```

SUBP 154
SUBP 155
SUBP 156
SUBP 157
SUBP 158
SUBP 159
SUBP 160
SUBP 161
SUBP 162
SUBP 163
SUBP 164
SUBP 165
SUBP 166
SUBP 167
SUBP 168
SUBP 169
SUBP 170
SUBP 171
SUBP 172
SUBP 173
SUBP 174
SUBP 175
SUBP 176
SUBP 177
SUBP 178
SUBP 179
SUBP 180
SUBP 181
SUBP 182
SUBP 183
SUBP 184
SUBP 185
SUBP 186
SUBP 187
SUBP 188
SUBP 189
SUBP 190
SUBP 191
SUBP 192
SUBP 193
SUBP 194
SUBP 195
SUBP 196
SUBP 197
SUBP 198
SUBP 199
SUBP 200
SUBP 201
SUBP 202
SUBP 203
SUBP 204
SUBP 205

SUBP 206
SUBP 207
SUBP 208
SUBP 209
SUBP 210
SUBP 211
SUBP 212
SUBP 213
SUBP 214
SUBP 215
SUBP 216
SUBP 217
SUBP 218
SUBP 219
SUBP 220
SUBP 221
SUBP 222
SUBP 223
SUBP 224
SUBP 225
SUBP 226
SUBP 227
SUBP 228
SUBP 229

```

ZO = ZREC + ZS(LS)
AY = YO
AZ = ZO
SGS = -SG(LS)
GO TO 30
CONTINUE
DPLR = DP
IF (ABS(YS(LS))) .LE. 0.001) GO TO 190
C LOWER LEFT SENDING POINT
IGO = 4
LHS = 1
SGS = SG(LS)
TL = -XLAM(J)
SL = TL/(SQRT(1.0+TL*TL))
YO = YREC + YS(LS)
AY = YO
GO TO 30
CONTINUE
180 DPLL = DP
190 CONTINUE
SUM(1) = DPUR + DPUL + FLNE*DPLR + FLNE*DPLL
SUM(2) = DPUR - DPUL + FLNE*DPLR - FLNE*DPLL
RETURN
END

```



```

230 IF (ABS(T1)-EPS) 230,240,240
231 ICHUZ=2
232 T1=0.
233 GO TO 300
240 ICHUZ=3
241 BETA2 = (1.-M*M)
242 BETA2 = SQRT(XO*XO+BETA2*R1*R1)
243 K1= KR*R1/BR
244 MU1= (M*BIGR-XO)/(BETA2*R1)
245 MU=ABS(MU1)
246 K2=K1*K1
247 IF (MU1) 310,320,330
248 ICHUZ=ICHUZ+3
249 GO TO 330
250 ICHUZ=ICHUZ+6
251 (N*C)**2 FOR N=1,11 AND C=.372 =
252
253 .138384 .553536 1.245456 2.214144
254 3.4596 4.981824 6.780816 8.856576
255 11.209104 13.8384 16.744464
256
257 (N*C) FOR N=1,12 AND 14,16,18,20,22 =
258
259 .744 1.116 1.488 1.86
260 2.604 2.976 3.348 3.72
261 4.464 5.208 5.952 6.696
262 8.184
263
264 A(N) FORN(=1,11) =
265
266 .24186198 -2.7918027 34.991070 -111.59196
267 271.43549 -305.75288 -41.18303 545.98537
268 -644.78155 328.72755 -64.279511
269 330 CONTINUE
270 EXARG = -0.372*MU
271 ** THE FOLLOWING TEST ON THE SIZE OF THE ARGUMENT TO EXP IS
272 NEEDED TO AVOID UNDERFLOW IN SUBPROGRAM EXP
273 IF (EXARG.GE.(-174.0)) GO TO 335
274 F = 0.0
275 GO TO 337
276 F = EXP(EXARG)
277 IF (F .LT. 1.E-30) F = 0.0
278 337 CONTINUE
279
280 C1 = .138334+K2
281 C2 = .553536+K2
282 C3 = 1.245456+K2
283 C4 = 2.214144+K2
284 C5 = 3.4596+K2
285 C6 = 4.981824+K2
286 C7 = 6.780816+K2
287 C8 = 8.856576+K2

```

TKER 54
 TKER 55
 TKER 56
 TKER 57
 TKER 58
 TKER 59
 TKER 60
 TKER 61
 TKER 62
 TKER 63
 TKER 64
 TKER 65
 TKER 66
 TKER 67
 TKER 68
 TKER 69
 TKER 70
 TKER 71
 TKER 72
 TKER 73
 TKER 74
 TKER 75
 TKER 76
 TKER 77
 TKER 78
 TKER 79
 TKER 80
 TKER 81
 TKER 82
 TKER 83
 TKER 84
 TKER 85
 TKER 86
 TKER 87
 TKER 88
 TKER 89
 TKER 90
 TKER 91
 TKER 92
 TKER 93
 TKER 94
 TKER 95
 TKER 96
 TKER 97
 TKER 98
 TKER 99
 TKER 100
 TKER 101
 TKER 102
 TKER 103
 TKER 104
 TKER 105

```

C9      = 11.209104+K2
C10     = 13.8384  +K2
C11     = 16.744464+K2

R1      = .24186198 / C1
R2      = -2.7918027 / C2
R3      = 24.991079 / C3
R4      = -111.59196 / C4
R5      = 271.43549 / C5
R6      = -305.75288 / C6
R7      = -41.18363 / C7
R8      = 545.98537 / C8
R9      = -644.78155 / C9
R10     = 328.72755 / C10
R11     = -64.279511 / C11

IF ( ICHUZ .LT. 4 )
  100R = .572*(R1+2.*R2 + 3.*R3 + 4.*R4 + 5.*R5 + 6.*R6 + 7.*R7 +
    8.*R8 + 9.*R9 + 10.*R10 + 11.*R11)
  100I = -K1*(R1+R2+R3+R4+R5+R6+R7+R8+R9+R10+R11)

340 GO TO (420,350,350,390,350,350,380,350,350), ICHUZ

350 Q1   = R1/ C1
Q2      = R2/ C2
Q3      = R3/ C3
Q4      = R4/ C4
Q5      = R5/ C5
Q6      = R6/ C6
Q7      = R7/ C7
Q8      = R8/ C8
Q9      = R9/ C9
Q10     = R10/ C10
Q11     = R11/ C11

GO TO (420,410,410,390,360,360,380,360,360), ICHUZ

360 J00R = Q1*(.138384-K2)+Q2*(.553536-K2)+Q3*(1.245456-K2)+Q4*
  1 (2.214144-K2)+Q5*(3.4596-K2)+Q6*(4.931824-K2)+Q7*(5.780816
  2 -K2)+Q8*(8.856576-K2)+Q9*(11.209104-K2)+Q10*(13.8384-K2)+
  3 Q11*(16.744464-K2)
  120R3= 2.*K1*I00I+K2*J00R

GO TO (420,410,410,390,410,390,380,370,370), ICHUZ

370 J00I = -K1*(.744*Q1+1.48*Q2+2.24*Q3+2.976*Q4+3.72*Q5+4.464*Q6+
  1 5.208*Q7+5.952*Q8+6.696*Q9+7.44*Q10+8.184*Q11)
  120I3= -K1*I00R+K2*J00I

IF ( ICHUZ .EQ. 8 )
  380 I10I = -K1*I00R
  390 I10R = 1.*K1*I00I

```

TKER 106
TKER 107
TKER 108
TKER 109
TKER 110
TKER 111
TKER 112
TKER 113
TKER 114
TKER 115
TKER 116
TKER 117
TKER 118
TKER 119
TKER 120
TKER 121
TKER 122
TKER 123
TKER 124
TKER 125
TKER 126
TKER 127
TKER 128
TKER 129
TKER 130
TKER 131
TKER 132
TKER 133
TKER 134
TKER 135
TKER 136
TKER 137
TKER 138
TKER 139
TKER 140
TKER 141
TKER 142
TKER 143
TKER 144
TKER 145
TKER 146
TKER 147
TKER 148
TKER 149
TKER 150
TKER 151
TKER 152
TKER 153
TKER 154
TKER 155
TKER 156
TKER 157

```

400 GO TO (420,410,410,420,410,410,500,500),ICHUZ
410 JOUR = E*(Q1*(.138384-K2+.372*MU*C1)+
1 E*(Q2*(.553536-K2+.744*MU*C2)+
2 E*(Q3*(1.245456-K2+1.116*MU*C3)+
3 E*(Q4*(2.214144-K2+1.188*MU*C4)+
4 E*(Q5*(3.4596-K2+1.36*MU*C5)+
5 E*(Q6*(4.981824-K2+2.232*MU*C6)+
6 E*(Q7*(6.780816-K2+2.504*MU*C7)+
7 E*(Q8*(8.856576-K2+2.976*MU*C8)+
8 E*(Q9*(11.209104-K2+3.348*MU*C9)+
9 E*(Q10*(13.8384-K2+3.72*MU*C10)+
A E*(Q11*(16.744464-K2+4.092*MU*C11))))))
JOUR = -K1*(E*(Q1*(.744*MU*C1) + F*(Q2*(1.488*MU*C2) +
1 E*(Q3*(2.232*MU*C3) + E*(Q4*(2.976*MU*C4) +
2 E*(Q5*(3.72*MU*C5) + E*(Q6*(4.464*MU*C6) +
3 E*(Q7*(5.208*MU*C7) + F*(Q8*(5.952*MU*C8) +
4 E*(Q9*(6.696*MU*C9) + E*(Q10*(7.44*MU*C10) +
5 E*(Q11*(8.184*MU*C11)))))))))
420 IOUR = .372*E*(R1+E*(2.*K2+E*(3.*R3+E*(4.*R4+E*(5.*R5+E*(6.*R6+
1 E*(7.*R7+E*(8.*R8+E*(9.*R9+E*(10.*R10+E*(11.*R11))))))
2 )))
IOU1 = -K1*(E*(R1+E*(R2+E*(R3+E*(R4+E*(R5+E*(R6+E*(R7+E*(R8+E*(R9
1 +E*(R10+E*(R11))))))))))
R1 = K1S
C6= K1*MU
C1= SIN(C6)
C2= COS(C6)
C3= SQRT(1.+MU*MU)
C4= MU/C3
C5= C4/(1.+MU*MU)
430 GO TO (430,440,430,430,440,430,500,500),ICHUZ
I1UR = C2*(1.-C4*K1*IOU1)-C1*K1*IOUR
I1UI = -C2*K1*IOUR-C1*(1.-C4*K1*IOU1)
440 GO TO (500,440,440,460,440,440,500,500),ICHUZ
I2UR3 = C2*(2.*(1.-C4)-C5*K1*IOU1+K2*JOUR)+C1*(C6*(1.-C4)-K1*IOUR
1 +K2*JOU1)
I2UI3 = C2*(C6*(1.-C4)-K1*IOUR+K2*JOUR)-C1*(2.*(1.-C4)-C5+K1*IOU1
1 +K2*JOUR)
GO TO (500,500,500,460,450,450,500,500),ICHUZ
450 I2UR3 = 2.0* I2OR3 - I2UR3
IF ( ICHUZ-5 )
460 CAR = 2.*I1OR-I1UR
I1UR= CAR
500 DK1R=0.
R1 = RIS
DK1I=0.
DK2R=0.

```

TKER 210
TKER 211
TKER 212
TKER 213
TKER 214
TKER 215
TKER 216
TKER 217
TKER 218
TKER 219
TKER 220
TKER 221
TKER 222
TKER 223
TKER 224
TKER 225
TKER 226
TKER 227
TKER 228
TKER 229
TKER 230
TKER 231
TKER 232
TKER 233
TKER 234
TKER 235
TKER 236
TKER 237
TKER 238
TKER 239
TKER 240
TKER 241
TKER 242
TKER 243
TKER 244
TKER 245
TKER 246
TKER 247
TKER 248
TKER 249
TKER 250
TKER 251
TKER 252
TKER 253

```

DK2I=0.
C3=K1*MU1
C1=COS(C3)
C2=SIN(C3)
C3=M*RI/BIGR
C4=SQR(1.+MU1*MU1)
C5=KR*X0/BR
C6=COS(C5)
C7=SIN(C5)
GO TO (530,540,530,530,540,530,510,520,510),ICHUZ
510 IUR=IUR
IUI=IUI
IF ( ICHUZ-7 )
520 I2UR3= I2UR3
I2UI3= I2UI3
IF ( ICHUZ-8 )
530 CK1R = IUR + C3*C1/C4
CK1I = IUI - C3*C2/C4
K10 = 1.0 + X0/BIGR
DK1R = CK1R*C6 + CK1I*C7
DK1I = CK1I*C6 - CK1R*C7
GO TO (900,540,540,900,540,540,900,540,540),ICHUZ
540 C8= ( BETA2*(R1/BIGR)**2 + (2.*MU1*C3)/(C4*C4) )*(-C3/C4)
C9= ( K1*C3)/(C3/C4)
CK2R = -I2UR3 + C8*C1 - C9*C2
CK2I = -I2UI3 - C9*C1 - C8*C2
K20 = -2.0 -X0*(2.0+BETA2*(R1/BIGR)**2)/BIGR
DK2R = CK2R*C6 + CK2I*C7
DK2I = CK2I*C6 - CK2R*C7
900 KKR = I1*DK1R + I2*DK2R
KKI = I1*DK1I + I2*DK2I
K1RI = I1 * DK1R
K1II = I1 * DK1I
K2RI2P = I2P * DK2R
K2II2P = I2P * DK2I
K10I1 = K10 * I1
K20I2P = K20 * I2P
905 CONTINUE
KDI1 = K1RI - K10I1*FLOAT(IND)
KDI1 = K1II
KD2R = K2RI2P-K20I2P*FLOAT(IND)
KD2I = K2II2P
RTURN
END

```

```

SUBROUTINE TVOR ( SL1, CL1, TL1, SL2, CL2, TL2, SG1, CG1,
*          CBAR, FMACH, KR, X01, X02, Y0, Z0, E, BETA,
*          IPRNT )
*          NORMALWASH AT A POINT (X,Y,Z) - OF A SURFACE DIHEDRAL
*          DUE TO A TRAPEZOIDAL UNSTEADY VORTEX RING OF UNIT
*          STRENGTH.
*          SUBROUTINES USED - SNPDF, IDP1, IDP2, FLLO
*          ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
*          SL1, CL1, TL1 SIN(LAMBDA-1), COS(LAMBDA-1), TAN(LAMBDA-1)
*          SL2, CL2, TL2 SIN(LAMBDA-2), COS(LAMBDA-2), TAN(LAMBDA-2)
*          SG1, CG1 SIN(GAMMA-S), COS(GAMMA-S)
*          X01, X02 X-X11, X-X12, Y-Y11, Y-Y12, Z-Z11, Z-Z12
*          Y0, Z0 Y - ETA, Z - ZETA
*          E BETA
*          CV CV
*          BR BR
*          FMACH FMACH
*          BRE BRE
*          BIM BIM
*          MACH NO.
*          REAL PART OF B (RETURNED)
*          IMAGINARY PART OF B (RETURNED)
*          ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
REAL KR, KD1, KD2
*          VARIABLES DIMENSIONED (2), FIRST WORD IS THE REAL PART
*          OF THE VALUE AND THE SECOND IS THE IMAGINARY PART
*          DIMENSION DK1 (2), DK2 (2), DK3 (2)
*          DATA NPOT /6/
*          DATA PI48 / 150.79644720 / CALCULATE BS
*          LCV = SL1
*          CL = CL1
*          TL = TL1
*          XO = X01
*          FE = F**2
*          TE = 2.0 * E
*          ASSIGN 50 TO ISNP

```

```

C      GO TO 1000          CALL SNPDF
50    BS = SL2             DIJ
      CL = CL2
      TL = TL2
      XO = XO2
      ASSIGN 100 TO ISNP
      CALL SNPDF
      GO TO 1000          RS - DIJ
      CALL SNPDF
      LIMITS FOR SMALL VALUES OF RADII
      FPS = 0.25 * FE
      IB = 0
      IC = 0
      FB = 1.0
      FC = 4.0
      FIRST CALC.
      DELTA-KD- II, IC, AND IO
      ETL1 = E * TL1
      ETL2 = E * TL2
      ESGS = E * SGS
      ECGS = E * CGS
      DX01 = X01 + ETL1
      DX02 = X02 + ETL2
      DY0 = Y0 + ECGS
      DZ0 = Z0 + ESGS
      ASSIGN 200 TO IFLLD
      CALCULATE R-1 SQUARED AND CALL FLLD IF LARGE ENOUGH
      R2 = DY0 ** 2 + DZ0 ** 2
      IF ( R2 .GE. EPS ) GO TO 2000
      IB = 1
      FC = 6.0
      FB = 0.0
      GO TO 230
      200 D0 220 I = 1, 2
      220 DK1(I) = KD1(I) / R2 + KD2(I) / R4
      230 DX01 = X01
      DX02 = X02
      DY0 = Y0
      DZ0 = Z0
      ASSIGN 300 TO IFLLD
      CALCULATE R-C SQUARED AND CALL FLLD IF LARGE ENOUGH
      R2 = DY0 ** 2 + DZ0 ** 2
      IF ( R2 .GE. EPS ) GO TO 2000
      IC = 1

```

TVOR 54
 TVOR 55
 TVOR 56
 TVOR 57
 TVOR 58
 TVOR 59
 TVOR 60
 TVOR 61
 TVOR 62
 TVOR 63
 TVOR 64
 TVOR 65
 TVOR 66
 TVOR 67
 TVOR 68
 TVOR 69
 TVOR 70
 TVOR 71
 TVOR 72
 TVCR 73
 TVOR 74
 TVOR 75
 TVOR 76
 TVOR 77
 TVOR 78
 TVCR 79
 TVOR 80
 TVOR 81
 TVOR 82
 TVOR 83
 TVOR 84
 TVOR 85
 TVOR 86
 TVOR 87
 TVOR 88
 TVOR 89
 TVOR 90
 TVOR 91
 TVOR 92
 TVOR 93
 TVOR 94
 TVOR 95
 TVCR 96
 TVCR 97
 TVOR 98
 TVOR 99
 TVOR 100
 TVOR 101
 TVOR 102
 TVOR 103
 TVCR 104
 TVOR 105

```

FC      = 0.0
FB      = 3.0
GO TO 330
300 DN 320 I = 1, 2
C
320 DKC(I) = KD1(I) / R2 + KD2(I) / R4
C
C      KD10 AND KD20
C      IS TOO SMALL
330 IF ( IH .NE. 0 ) GO TO 430
C      SKIP IF R-I IS TOO SMALL
DX01 = X01 - ETL1
DX02 = X02 - ETL2
DY0 = Y0 - ECGS
DZ0 = Z0 - ESGS
C
C      ASSIGN 400 TO IFLD
C      CALCULATE R-0 SQUARED AND CALL FLFD IF LARGE ENOUGH
R2 = DY0 ** 2 + DZ0 ** 2
IF ( R2 .GE. EPS ) GO TO 2000
FB = 0.0
FC = 6.0
IB = 1
GO TO 430
400 DN 420 I = 1, 2
C
420 DK0(I) = KD1(I) / R2 + KD2(I) / R4
C
430 COEF = 1.0 / PI48
BRE = BS / (TE * CV) - COEF * (FB * (DK1(1) + DK0(1)) + FC * DKC(1))
BIM = - COEF * (FB * (DK1(2) + DK0(2)) + FC * DKC(2))
C
C      RETURN
C * * * * *
C
1000 CALL SNPDF ( SL, CL, TL, SGS, CGS, DZ0, SGR, CGR, X0, Y0, Z0, E,
C      DIJ, BETA, CV )
C      GO TO ISNP, ( 50, 100 )
C
C
2000 CALL FLFD ( DX01, DX02, DY0, DZ0, SGR, CGR, SGS, CGS,
C      KR, CBAR, FMACH, E,
C      KD1(1), KD1(2), KD2(1), KD2(2) )
C      IF ( IPRNT .NE. 0 ) WRITE ( NPOI, 3000 ) KD1, KD2, "2
3000 FORMAT( 'OKD1=*,2E20.8,* KD2=*,2E20.8,* R2=*,E20.8' )
C
C      R4 = R2 * R2
C      GO TO IFLD, ( 200, 300, 400 )
C
END

```

TVOR 106
 TVOR 107
 TVOR 108
 TVOR 109
 TVOR 110
 TVOR 111
 TVOR 112
 TVOR 113
 TVOR 114
 TVOR 115
 TVOR 116
 TVOR 117
 TVOR 118
 TVOR 119
 TVOR 120
 TVOR 121
 TVOR 122
 TVOR 123
 TVOR 124
 TVOR 125
 TVOR 126
 TVOR 127
 TVOR 128
 TVOR 129
 TVOR 130
 TVOR 131
 TVOR 132
 TVOR 133
 TVOR 134
 TVOR 135
 TVOR 136
 TVOR 137
 TVOR 138
 TVOR 139
 TVOR 140
 TVOR 141
 TVOR 142
 TVOR 143
 TVOR 144
 TVOR 145
 TVOR 146
 TVOR 147
 TVOR 148
 TVOR 149
 TVOR 150
 TVOR 151
 TVOR 152
 TVOR 153
 TVOR 154

```

SUBROUTINE WANDWT(IPRINT, LINES, NOUT, NTOT, NBOX, NB, NDW, NTAPH,
1  NTPDH, NWT, NNTA,
2  RFFC, KR, H, DHDX, W, DW, COL
3  )
4
5  REAL
6  DIMENSION W(2, NTOT), DW(2, NTOT), COL(2, NTOT),
7  NTPDH, NWT, NNTA,
8  RFFC, KR, H, DHDX, W, DW, COL
9
10 FORMAT ( 1H1 / // )
11
12 KR
13 = NWT
14 NMODE = 0
15
16 READ (NTPDH) NSYM, NASYM
17 DO 15 N = 1, NTOT
18   DW(1, N) = 0.0
19   IF (IPRINT .NE. 0) WRITE (NOUT, 122)
20   IF (NB .EQ. 0) GO TO 70
21   REWIND NDW
22   CONTINUE
23   READ (NDW) ICOLB
24   IF (ICOLB .EQ. -1) GO TO 70
25   READ (NDW) COL
26   DO 30 I = 1, NTOT
27     DW(1, I) = COL(1, I)
28     DW(2, I) = COL(2, I)
29     W(1, I) = 0.0
30     W(2, I) = 0.0
31   CONTINUE
32   CONTINUE
33   READ (NTPDH) ICOLP
34   IF (ICOLP .EQ. -1) GO TO 150
35   READ (NTPDH) (H(J), J = 1, NBOX)
36   DO 90 I = 1, NBOX
37     W(1, I) = DHDX(I)
38     W(2, I) = 2.0*KR*H(I) / RFFC
39   CONTINUE
40   CONTINUE
41   DO 110 I = 1, NTOT
42     W(1, I) = -W(1, I) + DW(1, I)
43     W(2, I) = -W(2, I) + DW(2, I)
44   CONTINUE
45   WRITE (NWTAP) W
46
47   IF (IPRINT .EQ. 0) GO TO 140
48   IG2 = 0
49   LINES = LINES*3
50   IF (NB .NE. 0) LINES = LINES / 2
51   LPAGE = NTOT / LINES + 1
52   LPAGE = 1
53

```

```

DO 120 KG = 1, LPAGE
IG1 = IG2 + 1
IG2 = IG2 + LINE3
IG2 = NTOT
IF (IG2 .GT. NTOT) IG2 = NTOT
IF (NB .EQ. 0) GO TO 112
WRITE (NOUT,126) ICOLP, (1, DW(1,1), DW(2,1), I = IG1, IG2)
112 CONTINUE
WRITE (NOUT,130) ICOLP, (1, W(1,1), W(2,1), I = IG1, IG2)
IF (IG2 .GE. NTOT) GO TO 120
C
WRITE (NOUT,10)
120 CONTINUE
C
122 FORMAT ( 1H1 /// 37H *** SYMMETRIC MODES *** // )
124 FORMAT ( 1H1 /// 37H *** ANTISYMMETRIC MODES *** // )
126 FORMAT ( ( /// 8H COLUMN, 15, 12H OF --DW-- //(3(16,2E13.5)) )
130 FORMAT ( ( /// 8H COLUMN, 15, 12H UF --WT-- //(3(16,2E13.5)) )
C
140 CONTINUE
C
IF (NB .EQ. 0) GO TO 70
GO TO 20
150 CONTINUE
IF (IGO .EQ. 2 .OR. NASYM .EQ. 0) GO TO 160
IGO = IGO + 1
NWTAP = NWTAP + 1
IF (IPRINT .NE. 0) WRITE (NOUT, 124)
GO TO 140
160 CONTINUE
C
RETURN
END

```

WANDWT54
WANDWT55
WANDWT56
WANDWT57
WANDWT58
WANDWT59
WANDWT60
WANDWT61
WANDWT62
WANDWT63
WANDWT64
WANDWT65
WANDWT66
WANDWT67
WANDWT68
WANDWT69
WANDWT70
WANDWT71
WANDWT72
WANDWT73
WANDWT74
WANDWT75
WANDWT76
WANDWT77
WANDWT78
WANDWT79
WANDWT80
WANDWT81
WANDWT82
WANDWT83
WANDWT84
WANDWT85
WANDWT86

ZEROUT 2
 ZEROUT 3
 ZEROUT 4
 ZEROUT 5
 ZEROUT 6
 ZEROUT 7
 ZEROUT 8
 ZEROUT 9
 ZEROUT 10
 ZEROUT 11
 ZEROUT 12
 ZEROUT 13
 ZEROUT 14

```

C      SUBROUTINE ZEROUT( WORK, LENGTH, LOOP, ITAPE )
      COMPLEX WORK(LENGTH)

      DO 10 I = 1, LENGTH
      WORK(I) = 0.0
10    CONTINUE
      IF (ITAPE .EQ. 0) RETURN

C      DO 20 L = 1, LOOP
      WRITE (ITAPE) (WORK(I), I = 1, LENGTH)
20    CONTINUE
      RETURN
      END
  
```

```

SURROUTINE GENMS ( MS, MASS, LIMIT1, LIMIT2, ITM, IMS)
REAL MS, MASS
DIMENSION MS (ITM, ITM, 1), MASS (IMS, 1)
DO 10 I=1, IMS
  F(I)=1.0
  IMSP=IMS+1
  DO 20 I=IMSP, 23
    F(I)=0.
    DO 60 K = LIMIT1, LIMIT2
      DO 30 I = 1, ITM
        DO 30 J = 1, ITM
          MS (I, J, K) = 0.
          MS (1, 1, K) = MASS (1, K)*F(1)
          MS (2, 2, K) = MASS (1, K)*F(1)
          MS (3, 3, K) = MASS (1, K)*F(1)
          IF (ITM.LT.4) GO TO 40
          MS (2, 4, K) = -MASS (4, K)*F(4)
          MS (3, 4, K) = -MASS (3, K)*F(3)
          MS (4, 4, K) = -MASS (5, K)*F(5)
          IF (ITM.LT.5) GO TO 40
          MS (1, 5, K) = -MASS (4, K)*F(4)
          MS (3, 5, K) = -MASS (2, K)*F(2)
          MS (4, 5, K) = -MASS (8, K)*F(8)
          MS (5, 5, K) = -MASS (6, K)*F(6)
          IF (ITM.LT.6) GO TO 40
          MS (1, 6, K) = -MASS (3, K)*F(3)
          MS (2, 6, K) = -MASS (2, K)*F(2)
          MS (4, 6, K) = -MASS (10, K)*F(10)
          MS (5, 6, K) = -MASS (9, K)*F(9)
          MS (6, 6, K) = -MASS (7, K)*F(7)
          IF (ITM.LT.7) GO TO 40
          MS (1, 7, K) = -MASS (15, K)*F(15)
          MS (3, 7, K) = -MASS (11, K)*F(11)
          MS (4, 7, K) = -MASS (14, K)*F(14)
          MS (5, 7, K) = -MASS (12, K)*F(12)
          MS (6, 7, K) = -MASS (16, K)*F(16)
          MS (7, 7, K) = -MASS (13, K)*F(13)
          IF (ITM.LT.8) GO TO 40
          MS (1, 8, K) = -MASS (22, K)*F(22)
          MS (3, 8, K) = -MASS (17, K)*F(17)
          MS (4, 8, K) = -MASS (21, K)*F(21)
          MS (5, 8, K) = -MASS (18, K)*F(18)
          MS (6, 8, K) = -MASS (23, K)*F(23)
          MS (7, 8, K) = -MASS (19, K)*F(19)
          MS (8, 8, K) = -MASS (20, K)*F(20)
        CONTINUE
      DO 50 J = 1, ITM
        DO 50 I = J, ITM
          MS (I, J, K) = MS (J, I, K)
        CONTINUE
      RETURN
    END DO
  END DO
END DO

```

GENMS 54

END

INERTM 2
INERTM 3
INERTM 4
INERTM 5
INERTM 6
INERTM 7
INERTM 8
INERTM 9
INERTM10
INERTM11
INERTM12
INERTM13
INERTM14
INERTM15
INERTM16
INERTM17
INERTM18
INERTM19
INERTM20

```

C
SUBROUTINE INERTM(
1  NTL,NMS,EM,ELXIO,FLYIO,ELZIO
1  ,WR,PHIX,PHIY,PHIZ,EMS,AMODNO
2  ,CPXDPG,EMPHI,EMBAR,EMWR2,NOMOD,NSYM,NASYM)
C
  DIMENSION EM(1),ELXIO(1),ELYIO(1),ELZIO(1)
  DIMENSION WR(1),PHIX(1),PHIY(1),PHIZ(1),AMODNO(1)
  DIMENSION EMS(1)
  DIMENSION CPXDPG(1),EMPHI(1),EMBAR(1),EMWR2(1)
C
  CALL INPUTM(
1  NTL,NMS,EM,ELXIO,FLYIO,ELZIO
1  ,WR,PHIX,PHIY,PHIZ,AMODNO,CPXDPG)
C
  CALL LHS (EM,EMPHI,PHIX,PHIY,PHIZ,NMS,AMODNO,WR
1  ,EMS,EMBAR,EMWR2,NOMOD,NSYM,NASYM)
C
  RETURN
  END

```

```

SUBROUTINE INPUTM(NTI,NM
1  ,FM,ELXIO,ELZIO,WR,PHIX,PHIY,PHIZ,AMODNO,CPXDPG)
2  DIMENSION EM(1),ELXIO(1),ELZIO(1),ELZIO(1)
3  DIMENSION WR(1),PHIX(NM,1),PHIY(NM,1),PHIZ(NM,1)
4  DIMENSION AMODNO(20),CPXDPG(1)
5  DIMENSION MEMP(20)
6
7  COMMON/ZZZ/HEDR(48),NIN,NOUT,KROW,LINES
8
9  COMMON /XTE/XF(100)
10 EQUIVALENCE (XF(5),NMS)
11 EQUIVALENCE (XF(6),NSYM), (XF(7),NASYM), (XF(8),NMOD)
12 EQUIVALENCE (XF(80),ISECT), (XF(81),NDOF), (XF(82),IMS)
13 I, (XF(96),IPRNTM)
14
15 IF (ISECT.EQ.1) GO TO 200
16
17 IF (IPRNTM.EQ.0) GO TO 21
18 CALL HEADNG
19 WRITE (NOUT,1)
20 FORMAT(1H0,20X, MASS AND MASS POINT GEOMETRY /1H0)
21 CONTINUE
22
23 DO 90 I=1,NMS
24 READ (NTI,310) EM(I),ELXIO(I),ELZIO(I),ELZIO(I)
25 IF (IPRNTM.EQ.0) GO TO 22
26 WRITE (NOUT,480) (I,EM(I),ELXIO(I),ELZIO(I),ELZIO(I),I=1,NMS)
27 CONTINUE
28
29 DO 100 I=1,NMS
30 IF (FLYIO(I).EQ.0) EM(I)=0.50*EM(I)
31 CONTINUE
32
33 IF (IPRNTM.NE.0) CALL HEADNG
34
35 DO 190 M=1,NMOD
36
37 IF (IPRNTM.EQ.0) GO TO 23
38 WRITE (NOUT,2) M
39 FORMAT(1H0,20X, FREQUENCY AND INERTIAL MODE SHAPES FOR MODE ,14)
40 CONTINUE
41
42 READ (NTI,310) WR(M)
43 IF (IPRNTM.EQ.0) GO TO 24
44 LINES=LINES+4
45 WRITE (NOUT,520) M,WR(M),M,M,M
46 CONTINUE
47
48 DO 190 I=1,NMS
49 READ (NTI,310) PHIX(I,M),PHIY(I,M),PHIZ(I,M)
50 IF (IPRNTM.EQ.0) GO TO 190
51 LINES=LINES+1
52 IF (LINES.LT.KROW) GO TO 180
53

```

```

INPUTM 2
INPUTM 3
INPUTM 4
INPUTM 5
INPUTM 6
INPUTM 7
INPUTM 8
INPUTM 9
INPUTM 10
INPUTM 11
INPUTM 12
INPUTM 13
INPUTM 14
INPUTM 15
INPUTM 16
INPUTM 17
INPUTM 18
INPUTM 19
INPUTM 20
INPUTM 21
INPUTM 22
INPUTM 23
INPUTM 24
INPUTM 25
INPUTM 26
INPUTM 27
INPUTM 28
INPUTM 29
INPUTM 30
INPUTM 31
INPUTM 32
INPUTM 33
INPUTM 34
INPUTM 35
INPUTM 36
INPUTM 37
INPUTM 38
INPUTM 39
INPUTM 40
INPUTM 41
INPUTM 42
INPUTM 43
INPUTM 44
INPUTM 45
INPUTM 46
INPUTM 47
INPUTM 48
INPUTM 49
INPUTM 50
INPUTM 51
INPUTM 52
INPUTM 53

```

```

CALL HEADNG
WRITE (NOUT,2)M
WRITE (NOUT,520)M,WR(M),M,M,M
180 CONTINUE

C
WRITE (NOUT,530)I,PHIX(I,M),PHIY(I,M),PHIZ(I,M)
190 CONTINUE
GO TO 290

C
200 CALL INPUTS (NTI,NMS,NDOF,IMS,NOMOD,IPRNTM,ELXI(),WR,PHIX,EM)

C
290 CALL HEADNG
READ (NTI,310) (AMODNO(I),I=1,5)
READ (NTI,310) (AMODNO(I),I=6,9)
READ (NTI,310) (AMODNO(I),I=11,15)
READ (NTI,310) (AMODNO(I),I=16,20)
CC 300 I=1,20
300 MEMP(I) = AMODNO(I)
WRITE (NOUT,320) (MEMP(I),I=1,9)
320 FORMAT (1H0,20X, SYMMETRIC MODE
1 20X, RIGID BODY PLUNGE
1 20X, RIGID BODY PITCH
1 20X, RIGID BODY ROLL AND AFT
1 20X, FIRST SYMMETRIC ELASTIC
1 20X, LAST SYMMETRIC ELASTIC
1 20X, SYMMETRIC FLIGHT TRIM
1 20X, SYMMETRIC AIRCRAFT JIG
1 20X, FIRST DELETED SYMMETRIC
1 20X, LAST DELETED SYMMETRIC
WRITE (NOUT,321) (MEMP(I),I=11,20)
321 FORMAT (1H0,20X, ANTISYMMETRIC MODE LOCATION DEFINITION /1H0,
1 20X, RIGID BODY YAW
1 20X, RIGID BODY ROLL
1 20X, RIGID BODY LATERAL
1 20X, FIRST ANTISYM. ELASTIC
1 20X, LAST ANTISYM. ELASTIC
1 20X, ROLL TRIM
1 20X, YAW TRIM
1 20X, ANTISYM AIRCRAFT JIG
1 20X, FIRST DELETED ANTISYM
1 20X, LAST DELETED ANTISYM

C
25 CONTINUE

C
READ (NTI,310) (CPXDPG(I),I=1,NOMOD)
WRITE (NOUT,540) (I,CPXDPG(I),I=1,NOMOD)

C
RETURN

C
310 FORMAT (6F12.0)
320 FORMAT (2X,3H 1,5X,5HEM(I),8X,8HELXI(I),7X,8HELYI(I),7X,8HELZ(I)
430 1(I)/(15,1X,F13.6,2X,F13.6,2X,F13.6,2X,F13.6))

```

INPUTM54
 INPUTM55
 INPUTM56
 INPUTM57
 INPUTM58
 INPUTM59
 INPUTM60
 INPUTM61
 INPUTM62
 INPUTM63
 INPUTM64
 INPUTM65
 INPUTM66
 INPUTM67
 INPUTM68
 INPUTM69
 INPUTM70
 INPUTM71
 INPUTM72
 INPUTM73
 INPUTM74
 INPUTM75
 INPUTM76
 INPUTM77
 INPUTM78
 INPUTM79
 INPUTM80
 INPUTM81
 INPUTM82
 INPUTM83
 INPUTM84
 INPUTM85
 INPUTM86
 INPUTM87
 INPUTM88
 INPUTM89
 INPUTM90
 INPUTM91
 INPUTM92
 INPUTM93
 INPUTM94
 INPUTM95
 INPUTM96
 INPUTM97
 INPUTM98
 INPUTM99
 INPUT100
 INPUT101
 INPUT102
 INPUT103
 INPUT104
 INPUT105

```

520  FCRMAT (4H WR(I2,4H) = F11.4/2X,3H I,3X,8H PHIX(I,I2,1H),5X,7H PHI INPUT106
      1Y(I,I2,1H),5X,7H PHI2(I,I2,1H) INPUT107
530  FCRMAT (I5,3(2X,E13.6)) INPUT108
      540 FCRMAT (I10,3X1H1.4X9HCPXDPG(I)/(1X14,E13.6)) INPUT109
      END INPUT110

```

```

SUBROUTINE INPUTS (NTI,NMS,NDOF,IMS,NOMOD,IPK,EL,WR,PHI,EM)
DIMENSION EL(NMS,1), WR(1), PHI(NMS,NOMOD,1), EM(IMS,1)
COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINES
DIMENSION DEFL(8)
DATA DEFL/3H F,3H L,3H H,5H THETA,5H ALPHA,4H PSI,5H BETA,5H DELTA/
310 FORMAT (6F12.0)
320 FORMAT (1H0,20X,*MODESHAPE FOR MODE*,14,4X5HFREQ=,E11.4
1 / 5H SECT,6X,A5,8(9XA5) )
321 FORMAT (15,9E14.6)
DO 110 I=1,NMS
READ (NTI,310) EL(I,1), EL(I,2), EL(I,3)
IF (IPR.EQ.0) GO TO 110
WRITE (NOUT,321) I, (EL(I,J),J=1,3)
110 CONTINUE
C
READ (NTI,310) (WR(I),I=1,NOMOD)
DO 130 K=1,NOMOD
LINES = KROW
DO 120 J=1,NMS
READ (NTI,310) (PHI(J,K,I),I=1,NDOF)
IF (IPR.EQ.0) GO TO 120
IF (LINES.LT.KROW) GO TO 115
CALL HEADNG
WRITE (NOUT,320) K, WR(K), (DEFL(I),I=1,NDOF)
LINES = LINES+4
115 WRITE (NOUT,321) J, (PHI(J,K,I),I=1,NDOF)
LINES = LINES+1
120 CONTINUE
130 CONTINUE
C
DO 140 J=1,NMS
READ (NTI,310) (EM(I,J),I=1,4)
READ (NTI,310) (EM(I,J),I=5,10)
IF (NDOF.LT.7) GO TO 140
READ (NTI,310) (EM(I,J),I=11,16)
IF (NDOF.LT.8) GO TO 140
READ (NTI,310) (EM(I,J),I=17,20)
READ (NTI,310) (EM(I,J),I=21,23)
140 CONTINUE
CALL PRINT (FM,IMS,NMS,IMS,1,16HMASS PROPERTIES ,4)
C
DO 150 J=1,NMS
IF (EL(J,2).NE.0) GO TO 150
DO 145 I=1,IMS
145 EM(I,J) = 0.5*EM(I,J)
150 CONTINUE
C

```

INPUTS 2
INPUTS 3
INPUTS 4
INPUTS 5
INPUTS 6
INPUTS 7
INPUTS 8
INPUTS 9
INPUTS 10
INPUTS 11
INPUTS 12
INPUTS 13
INPUTS 14
INPUTS 15
INPUTS 16
INPUTS 17
INPUTS 18
INPUTS 19
INPUTS 20
INPUTS 21
INPUTS 22
INPUTS 23
INPUTS 24
INPUTS 25
INPUTS 26
INPUTS 27
INPUTS 28
INPUTS 29
INPUTS 30
INPUTS 31
INPUTS 32
INPUTS 33
INPUTS 34
INPUTS 35
INPUTS 36
INPUTS 37
INPUTS 38
INPUTS 39
INPUTS 40
INPUTS 41
INPUTS 42
INPUTS 43
INPUTS 44
INPUTS 45
INPUTS 46
INPUTS 47
INPUTS 48
INPUTS 49
INPUTS 50
INPUTS 51
INPUTS 52
INPUTS 53

INPUTS4
INPUTS5

RETURN
END

```

SUBROUTINE LHS (FM,EMPHI,PHIX,PHIY,PHIZ,NMS
1 , AMODNG, FREQ, EMS, CLHS, EMWR2, NEQ, NSYM, NASYM)
COMMON/ZZZ/CASE(48), NIN,NDUT,KROW,LINES,IPRNT,NER
COMMON/XTF/NF(100)
EQUIVALENCE (NF(48),IDIMUL)
EQUIVALENCE (NF(80),ISECT), (NF(81),NDOF), (NF(82),IMS)
1 , (NF(96),IPRNTM)
DIMENSION EM(1), EMPHI(3,NMS,1), EMS(1)
DIMENSION PHIX(NMS,1),PHIY(NMS,1),PHIZ(NMS,1)
DIMENSION AMODNG(20)
DIMENSION CLHS (NEQ,NEQ),FREQ(1),EMWR2(1)
DO 10 I=1,NEQ
DO 10 J=1,NEQ
CLHS(I,J)=0.0
10 CONTINUE
IF (ISECT.EQ.1) GO TO 50
IF (IDIMUL.FQ.0)GO TO 40
DO 60 IM=1,NMS
DO 30 J=1,NEQ
EMPHI(1,IM,J) = EM(IM) * PHIX(IM,J)
EMPHI(2,IM,J) = EM(IM) * PHIY(IM,J)
EMPHI(3,IM,J) = EM(IM) * PHIZ(IM,J)
DO 30 I=1,NEQ
CLHS(I,J) = CLHS(I,J) + PHIX(IM,I) * EMPHI(1,IM,J)
+ PHIY(IM,I) * EMPHI(2,IM,J) + PHIZ(IM,I) * EMPHI(3,IM,J)
1 CONTINUE
GO TO 55
40 CONTINUE
DO 45 I=1,NEQ
DO 45 J=1,NEQ
DO 45 K=1,NMS
CLHS(I,J)=CLHS(I,J)+PHIX(K,I)*PHIX(K,J)*EM(K)
+PHIY(K,I)*PHIY(K,J)*EM(K)
+PHIZ(K,I)*PHIZ(K,J)*EM(K)
1 CONTINUE
GO TO 55
45 CONTINUE
50 CALL GENMS (EMS,FM,1,NMS,NDOF,IMS)
IF (IDIMUL.FQ.0) CALL TMST (CLHS,EMS,PHIX,NEQ,NDOF,1,NMS)
IF (IDIMUL.NE.0) CALL TMSTL (NDOF,1,NMS,NEQ,EMS,CLHS,EMPHI,PHIX)
55 CONTINUE
IF (NASYM.EQ.0.OR.NASYM.EQ.0)GO TO 67
NOW ZERO OUT THOSE GM TERMS THAT S/N BE THERE (1/2 A/C ANAL)

```


LHS 106
LHS 107
LHS 108
LHS 109
LHS 110

CALL PRNT(EMWP2,NEQ,1,NEQ,1,
1 28HGENERALIZED STIFFNESS MATRIX ,7)
RETURN
END

C

```

SUBROUTINE TMST ( EMBAR, MS, TM, NMODES, ITM, NFTMB, NLTM)
REAL MS
DIMENSION MS (ITM, ITM, I, I), TM(NLTM, NMODES, I)
DIMENSION EMBAR(NMODES, NMODES), TEMP(8)
DO 137 K = 1, NMODES
DO 133 J = 1, NMODES
DO 131 I = 1, ITM
TEMP (I) = 0.0
DO 131 L = 1, ITM
TEMP (I) = MS (I, L, K) * TM (K, J, L) + TEMP (I)
DO 133 I = J, NMODES
DO 133 L = 1, ITM
EMBAR (I, J) = TM (K, I, L) * TEMP (L) + EMBAR (I, J)
DO 137 J = 1, NMODES
DO 137 I = J, NMODES
EMBAR (J, I) = EMBAR (I, J)
RETURN
END

```

```

TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST
TMST

```

```

SUBROUTINE TMSTL(IIM,NFTMB,NLTMB,NM,MS,EMBAR,EMPHI,IM)
REAL MS
DIMENSION MS(ITM,ITM,NLTMB)
DIMENSION EMBAR(NM,NM)
DIMENSION EMPHI(ITM,NLTMB,NM)
DIMENSION TM(NLTMB,NM,ITM)
DO 10 J=1,ITM
DO 10 K=1,NM
DO 10 I=NFTMB,NLTMB
EMPHI(J,I,K)=0.
DO 10 L=1,ITM
10 EMPHI(J,I,K)=EMPHI(J,I,K)+MS(J,L,I)*TM(I,K,L)
DO 30 I=1,NM
DO 30 J=1,NM
DO 20 L=1,ITM
DO 20 K=NFTMB,NLTMB
20 EMBAR(I,J)=EMBAR(I,J)+TM(K,I,L)*FMPHI(L,K,J)
30 CONTINUE
RETURN
END

```

```

TMSTL 2
TMSTL 3
TMSTL 4
TMSTL 5
TMSTL 6
TMSTL 7
TMSTL 8
TMSTL 9
TMSTL 10
TMSTL 11
TMSTL 12
TMSTL 13
TMSTL 14
TMSTL 15
TMSTL 16
TMSTL 17
TMSTL 18
TMSTL 19
TMSTL 20
TMSTL 21

```

```

SUBROUTINE LOAD(
1  NTI, ELXIO, FLYIO, FLZIO, FMPI, NDOF, VMS, NSYM, NASYM,
1  PHIX, PHIZ, NENG, NENGM)
2
3  DIMENSION FLXIO(1), ELYIO(1), FLZIO(1)
4  DIMENSION FMPI(NDOF,NMS,1)
5  DIMENSION PHIX(NMS,1), PHIZ(NMS,1), NENGM(1)
6  DIMENSION IHD(50)
7
8  COMMON NAA,A(1)
9  COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINES,IPRNT,NER
10 COMMON /XTE/XF(100)
11 EQUIVALENCE (XF(11),IDENT)
12 EQUIVALENCE (XF(31),NBEAMS), (XF(32),NINTLD), (XF(33),NSTRSS)
13 , (XF(34),NMGRP), (XF(35),NABGRP), (XF(36),NSBGRP)
14 , (XF(37),NBOXES), (XF(38),NAERSB)
15 , (XF(41),NK), (XF(42),NG)
16 , (XF(43),NBOX), (XF(44),NSBETO)
17 , (XF(24),KPRLDS), (XF(8),NMOD)
18 , (XF(80),ISECT)
19 COMMON /DDTBLS/DDTBL(20,10)
20 INTEGER DDTBL
21 DATA COREKO/8HUNITLOAD/
22
23 999 FORMAT(1H0,20X,I10, WORDS OF CORE REQ FOR STEP +++,A10,+++)
24
25  NM = NSYM + NASYM
26
27  L1 = NAA
28  L2 = L1 + NBEAMS*7
29  L3 = L2 + NINTLD*8
30  L4 = L3 + 3*NMGRP
31  L5 = L4 + 3*NABGRP
32  L6 = L5 + 3*NSBGRP
33  L7 = L6 + NSTRSS*NINTLD
34
35  CALL INPUTL (NTI,A(11),NBEAMS,A(12),NINTLD,A(16),NSTRSS
36  , A(13),NMGRP,A(14),NABGRP,A(15),NSBGRP,NMASS,NBOXES,NAERSB
37  , A(17),NENG,NENGM)
38
39  IHD(2) = IDENT
40  IHD(3) = NINTLD
41  IHD(4) = NSTRSS
42  IHD(5) = NENG
43  IHD(6) = NSYM
44  IHD(7) = NASYM
45  IHD(8) = NK
46  IHD(9) = NABGRP
47  IHD(10) = NSBGRP
48  IHD(11) = NBOXES
49  IHD(12) = NAERSB
50  NC 100 I=13,50
51
52
53

```



```

C      CALL HEADNG
C      DC 200 L=1,NMODC
C      WRITE(NOUT,190)
C      FORMAT(1H,20X, MASS TIMES MODES, MODE,14)
C      CALL PRNT (FMPI(1,1,L), NDOF, NMS, NDOF,1,16HMPH((NDOF,NMS)
C      ,3)
C      CONTINUE
C      210 CALL MSSPHI (NMGRP, A(L3), NINTLD
C      1, A(L12), A(L13), A(L14), A(L141), A(L142), A(L143)
C      2, NDOF,NMS,FMPI,NM,A(L15), A(L91), A(L92), NSYM,NASYM )
C      WRITE PIQ ON NTU
C      CALL WPUNIT (6.0,A(L15),NOUT,NER)
C      LENGTH = NINTLD*NARGRP*NBOXES
C      L13 = L12 + LENGTH
C      LENGTH = NINTLD*NSBGRP*NAERSB
C      L14 = L13 + LENGTH
C      L15 = L14 + LENGTH
C      L16 = L15 + NBOXES
C      L17 = L16 + NBOXES
C      L18 = L17 + NBOXES
C      CALL RDAERO (4HGE01,0,A(L18),NOUT,NER)
C      IF (NER.NF.0) GO TO 1000
C      L19 = L18 + NBOX
C      L20 = L19 + NBOX
C      L21 = L20 + NBOX
C      L22 = L21 + NBOX
C      L23 = L22 + NBOX
C      L24 = L23 + NBOX
C      WRITE (NOUT,999) L24,CORERQ
C      CALL GEOMAB (NARGRP,A(L4),NBEAMS,A(L9),A(L8),NINTLD,A(L2)
C      1, A(L12),A(L15),A(L16),A(L17),A(L11),A(L10)
C      2, A(L18),A(L19),A(L20),A(L21),A(L22),A(L23) )
C      L16 = L15 + NAERSB
C      L17 = L16 + NAERSB
C      L18 = L17 + NAERSB
C      CALL RDAERO (4HGE0S,0,A(L18),NOUT,NER)
C      IF (NER.NE.0) GO TO 1000
C      L19 = L18 + NSBETO
C      L20 = L19 + NSBETO
C      L21 = L20 + NSBETO
C      L22 = L21 + NSBETO
C      L23 = L22 + NSBETO
C      L24 = L23 + NSBETO
C      WRITE (NOUT,999) L24,CORERQ

```

```

LOAD 106
LOAD 107
LOAD 108
LOAD 109
LOAD 110
LOAD 111
LOAD 112
LOAD 113
LOAD 114
LOAD 115
LOAD 116
LOAD 117
LOAD 118
LOAD 119
LOAD 120
LOAD 121
LOAD 122
LOAD 123
LOAD 124
LOAD 125
LOAD 126
LOAD 127
LOAD 128
LOAD 129
LOAD 130
LOAD 131
LOAD 132
LOAD 133
LOAD 134
LOAD 135
LOAD 136
LOAD 137
LOAD 138
LOAD 139
LOAD 140
LOAD 141
LOAD 142
LOAD 143
LOAD 144
LOAD 145
LOAD 146
LOAD 147
LOAD 148
LOAD 149
LOAD 150
LOAD 151
LOAD 152
LOAD 153
LOAD 154
LOAD 155
LOAD 156
LOAD 157

```

```

C      CALL GEOMSB (NSRGRP,A(L5),NBEAMS,A(L9),A(L8),NINTLD,A(L2)
1      , A(L13),A(L14),A(L15),A(L16),A(L17),A(L11),A(L10)
2      , A(L18),A(L19),A(L20),A(L21),A(L22),A(L23) )
C      MXROW = MAX0(NBOX,NSBETC)
      MXCOL=MAX0(NSYM,NASYM)
      LI6 = LI5 + 2*MXROW*MXCOL
      LI7 = LI6 + 2*NINTLD*MXCOL
C      WRITE(NDUT,999)LI7,CORERO
      CALL AROLOD (NINTLD,A(L16),NSYM,NASYM,NK,NABGRP
1      , A(L4),NSBGRP,A(L5),A(L12),A(L13),A(L14),A(L15),
2      , A(L91),A(L92))
C      1000 RETURN
      END
LOAD 158
LOAD 159
LOAD 160
LOAD 161
LOAD 162
LOAD 163
LOAD 164
LOAD 165
LOAD 166
LOAD 167
LOAD 168
LOAD 169
LOAD 170
LOAD 171
LOAD 172
LOAD 173
LOAD 174

```

ABAMLT 2
ABAMLT 3
ABAMLT 4
ABAMLT 5
ABAMLT 6
ABAMLT 7
ABAMLT 8
ABAMLT 9
ABAMLT 10
ABAMLT 11
ABAMLT 12
ABAMLT 13
ABAMLT 14
ABAMLT 15
ABAMLT 16
ABAMLT 17
ABAMLT 18
ABAMLT 19
ABAMLT 20
ABAMLT 21
ABAMLT 22
ABAMLT 23
ABAMLT 24
ABAMLT 25
ABAMLT 26
ABAMLT 27
ABAMLT 28
ABAMLT 29
ABAMLT 30
ABAMLT 31
ABAMLT 32
ABAMLT 33
ABAMLT 34
ABAMLT 35
ABAMLT 36
ABAMLT 37
ABAMLT 38
ABAMLT 39
ABAMLT 40

```

SUBROUTINE ABAMLT(LOADCD,PINTP,TLAMM,DX,DY,DZ,GAMZ,GAMY)
  DIMENSION TLAMM(3,3)
  GO TO (501,502,503,504,505,506),LOADCD

501 AL=-TLAMM(1,1)*DZ+TLAMM(1,3)*DX
    AH=TLAMM(1,1)*DY-TLAMM(1,2)*DX
    GO TO 510

502 AL=-TLAMM(2,1)*DZ+TLAMM(2,3)*DX
    AH=TLAMM(2,1)*DY-TLAMM(2,2)*DX
    GO TO 510

503 AL=-TLAMM(3,1)*DZ+TLAMM(3,3)*DX
    AH=TLAMM(3,1)*DY-TLAMM(3,2)*DX
    GO TO 510

504 AL=TLAMM(3,2)
    AH=TLAMM(3,3)
    GO TO 510

505 AL=TLAMM(2,2)
    AH=TLAMM(2,3)
    GO TO 510

506 AL=TLAMM(1,2)
    AH=TLAMM(1,3)

510 CONTINUE
    NOW ROTATE TO AERO LOAD SYSTEM
    PINTP=-AL*GAMY+AH*GAMZ

    RETURN
  END

```

AROL002
AROL003
AROL004
AROL005
AROL006
AROL007
AROL008
AROL009
AROL010
AROL011
AROL012
AROL013
AROL014
AROL015
AROL016
AROL017
AROL018
AROL019
AROL020
AROL021
AROL022
AROL023
AROL024
AROL025
AROL026
AROL027
AROL028
AROL029
AROL030
AROL031
AROL032
AROL033
AROL034
AROL035
AROL036
AROL037
AROL038
AROL039
AROL040
AROL041
AROL042
AROL043
AROL044
AROL045
AROL046
AROL047
AROL048
AROL049
AROL050
AROL051
AROL052
AROL053

```

C      READ DZOS OR DZCA INTO FPH
C      IA = IA+1
C      CALL RDAERO (AERO(IA),K,FPH,NOUT,NER)
C      CALL AROPHA(NSBGRP,NFNLSB,NINTLD,PINTZ,NSBETO,FPH,NMT,PAQ)
C      IF(KPRCHK,EQ.0)GO TO 410
C      CALL PRNT(FPH,NSBETO,NMT,NSBETO,2,12HDZOS OR DZOA ,3)
C      IF(NLOOP,EQ.1)WRITE(NOUT,1)K
C      IF(NLOOP.GT.1)WRITE(NOUT,2)K
C      CALL PRNT( PAQ,NINTLD,NMT,NINTLD,2,12HPAQ5 OR PAQA ,3)
C      410 CONTINUE
C      READ DYOS OR DYCA INTO FPH
C      IA = IA+1
C      CALL RDAERO (AERO(IA),K,FPH,NOUT,NER)
C      CALL AROPHA(NSBGRP,NFNLSB,NINTLD,PINTY,NSBETO,FPH,NMT,PAQ)
C      IF(KPRCHK,EQ.0)GO TO 420
C      CALL PRNT(FPH,NSBETO,NMT,NSBETO,2,12HDYOS OR DYOA ,3)
C      IF(NLOOP,EQ.1)WRITE(NOUT,1)K
C      IF(NLOOP.GT.1)WRITE(NOUT,2)K
C      CALL PRNT( PAQ,NINTLD,NMT,NINTLD,2,12HPAQ5 OR PAQA ,3)
C      420 CONTINUE
C      NEW CLEAN UP FOR C/L INTGD LOADS
C      DO 325 L=1,NINTLD
C      IF(NLOOP,EQ.1)SCODE=SYMCOD(L)
C      IF(NLOOP,EQ.2)SCODE=ASMCOD(L)
C      DO 325 M=1,NMT
C      PAQ(1,L,M)=PAQ(1,L,M)*SCODE
C      PAQ(2,L,M)=PAQ(2,L,M)*SCODE
C      IF(KPRCHK.NE.0)CALL PRNT(PAQ,NINTLD,NMT,NINTLD,2,
C      1 12HPAQ5 OR PAQA MODED FOR C/L LOADS ,8)
C      NEW SAVE PAQS CR PAQA ON NTU
C      IP = NLOOP+6
C      CALL WRUNIT (IR,K,PAQ,NOUT,NER)
C      LOOP BACK FOR ANTISYM AERO IF NOT COMPLETE
C      NLOOP=NLOOP+1
C      IF(NLOOP.GT.2)GO TO 1000

```

AROLO054
 AROLO055
 AROLO056
 AROLO057
 AROLO058
 AROLO059
 AROLO060
 AROLO061
 AROLO062
 AROLO063
 AROLO064
 AROLO065
 AROLO066
 AROLO067
 AROLO068
 AROLO069
 AROLO070
 AROLO071
 AROLO072
 AROLO073
 AROLO074
 AROLO075
 AROLO076
 AROLO077
 AROLO078
 AROLO079
 AROLO080
 AROLO081
 AROLO082
 AROLO083
 AROLO084
 AROLO085
 AROLO086
 AROLO087
 AROLO088
 AROLO089
 AROLO090
 AROLO091
 AROLO092
 AROLO093
 AROLO094
 AROLO095
 AROLO096
 AROLO097
 AROLO098
 AROLO099
 AROLO100
 AROLO101
 AROLO102
 AROLO103
 AROLO104
 AROLO105

AROL0106
AROL0107
AROL0108
AROL0109
AROL0110
AROL0111
AROL0112
AROL0113
AROL0114
AROL0115
AROL0116
AROL0117
AROL0118
AROL0119
AROL0120
AROL0121
AROL0122
AROL0123
AROL0124
AROL0125
AROL0126
AROL0127
AROL0128
AROL0129
AROL0130

```

IF (NASYM.EQ.0) GO TO 1000
IA = 4
NMT=NASYM
GC TO 100
C 1000 CONTINUE
C
C WRITE SYMCOU AND ASMCOU ON UNIT LOAD TAPE
C
C CALL WRUNIT(9,0,SYMCCO,NOUT,NER)
C CALL WRUNIT(10,0,ASMCCO,NOUT,NER)
C
C NOW SAVE NFNLAB AND PINTP ON NTU
C
C CALL WRUNIT (11,0,NFNLAB,NOUT,NER)
C CALL WRUNIT (12,0,PINTP,NOUT,NER)
C
C NOW SAVE NFNLBR AND PINTY AND PINTZ ON NTU
C
C CALL WRUNIT (13,0,NFNLBR,NOUT,NER)
C CALL WRUNIT (14,0,PINTZ,NOUT,NER)
C CALL WRUNIT (15,0,PINTY,NOUT,NER)
C
C RETURN
C END

```

AROPHA 2
AROPHA 3
AROPHA 4
AROPHA 5
AROPHA 6
AROPHA 7
AROPHA 8
AROPHA 9
AROPHA10
AROPHA11
AROPHA12
AROPHA13
AROPHA14
AROPHA15
AROPHA16
AROPHA17
AROPHA18
AROPHA19
AROPHA20
AROPHA21
AROPHA22

SUBROUTINE AROPHA(NGRP,NFNL,NINTLD,PINT,NROW,FPH,NCOL,P)

DIMENSION P(2,NINTLD,1)

DIMENSION PINT(NINTLD,NGRP,1)

DIMENSION FPH(2,NROW,1)

DIMENSION NFNL(3,1)

DC 100 M=1,NINTLD

DC 100 N=1,NCOL

DC 100 L=1,NGFP

NAFRO=NFAL(2,L)-NFNL(1,L)+1

DC 100 J=1,NAFRO

LFC=NFNL(1,L)+J-1

P(1,M,N)=P(1,M,N)+PINT(M,L,J)*FPH(1,LOC,N)

P(2,M,N)=P(2,M,N)+PINT(M,L,J)*FPH(2,LOC,N)

CONTINUE

RETURN

END


```

DZ=BEAMGM(I,6)-BEAMGM(I,3)
IF (ABS(DX).LT.EPS)DX=0.
IF (ABS(DY).LT.EPS)DY=0.
IF (ABS(DZ).LT.EPS)DZ=0.
AL=SQRT(DX**2+DY**2+DZ**2)
DDL=SQRT(DY**2+DZ**2)
CCSSWP=DDL/AL
SINSWP=DX/AL
IF (DDL.LT.EPS)GO TO 105
COSDIH=DY/DDL
SINDIH=DZ/DDL
GO TO 106

C
C 105
COSDIH=0.
SINDIH=1.
C 106
CONTINUE

FORM DIRECTION COSINE MATRIX, XBEAM=R*XAAS

R11=CCSSWP
R12=-SINSWP*COSDIH
R13=-SINSWP*SINDIH
R21=SINSWP
R22=CCSSWP*COSDIH
R23=CCSSWP*SINDIH
R31=0.
R32=-SINDIH
R33=COSDIH

FORM DIR.COS. FOR YBEAM IN TERMS OF X,Y,Z AAS

TLAMY(1,1)=R21
TLAMY(2,1)=R22
TLAMY(3,1)=R23

FIND ORIGIN OF BEAM IN AAS SYSTEM

TLAMY(4,1)=- (R21*BEAMGM(I,1)+R22*BEAMGM(I,2)+R23*BEAMGM(I,3))
I=BEAMGM(I,7)

FORM INTG D LOAD DIR.COS IN TERMS OF AAS LOADS, BY BEAM COMP. NO.
COMP. SEQ. IS
1=WING OR HORIZONTAL TAIL
2=FUSELAGE
3=VERTICAL TAIL
4=WING PODS
5=FUSELAGE PODS
6=VERTICAL STAB. PODS

TLAMM(1,1,I) = P11

```

BEAM 54
 BEAM 55
 BEAM 56
 BEAM 57
 BEAM 58
 BEAM 59
 BEAM 60
 BEAM 61
 BEAM 62
 BEAM 63
 BEAM 64
 BEAM 65
 BEAM 66
 BEAM 67
 BEAM 68
 BEAM 69
 BEAM 70
 BEAM 71
 BEAM 72
 BEAM 73
 BEAM 74
 BEAM 75
 BEAM 76
 BEAM 77
 BEAM 78
 BEAM 79
 BEAM 80
 BEAM 81
 BEAM 82
 BEAM 83
 BEAM 84
 BEAM 85
 BEAM 86
 BEAM 87
 BEAM 88
 BEAM 89
 BEAM 90
 BEAM 91
 BEAM 92
 BEAM 93
 BEAM 94
 BEAM 95
 BEAM 96
 BEAM 97
 BEAM 98
 BEAM 99
 BEAM 100
 BEAM 101
 BEAM 102
 BEAM 103
 BEAM 104
 BEAM 105

BEAM 106
BEAM 107
BEAM 108
BEAM 109
BEAM 110
BEAM 111
BEAM 112
BEAM 113
BEAM 114
BEAM 115
BEAM 116
BEAM 117
BEAM 118
BEAM 119
BEAM 120
BEAM 121
BEAM 122
BEAM 123
BEAM 124
BEAM 125
BEAM 126
BEAM 127
BEAM 128
BEAM 129
BEAM 130

```

      TLAMM(1,2,I) = R12
      TLAMM(1,3,I) = R13
      TLAMM(2,1,I) = R21
      TLAMM(2,2,I) = R22
      TLAMM(2,3,I) = R23
      TLAMM(3,1,I) = R31
      TLAMM(3,2,I) = R32
      TLAMM(3,3,I) = R33
      K=NRCON(I)
      WRITE(INOUT,2)I,COSEDIH,SINDIH,COSSWP,SINSWP,
1 (NBSEQ(I,J),J=1,K)
2 FORMAT (1H0,I4,4F10.6,4X,20I3/10(1H ,48X,20I3/1))
C 200 CONTINUE
C
      CALL PRNT (TLAMY,4,NBEAMS,4,1,5HTLAMY,2)
      DC 250 K=1,NBEAMS
      WRITE (6,240) K
240 FORMAT (1H0,6HNBEAM=,I3)
      LINES = LINES+2
250 CALL PRNT (TLAMM(1,1,K),3,3,3,1,5HTLAMM,2)
      RETURN
      END

```



```

C 100 J=1,NBOXES
C LOC=NFNLBH(1,L)+J-1
C REXIN(J)=(X12(LOC)+X11(LOC))/2.
C RBYIN(J)=(ETA2(LOC)+ETA1(LOC))/2.
C RBYIO(J)=(ZETA2(LOC)+ZETA1(LOC))/2.
C
C 7
C 100
C 100 CONTINUE
C IF(KPRLOS.EQ.0)GO TO 105
C WRITE(NOUT,6)
C 6
C 105
C 105 CONTINUE
C 105 FIND ALL BEAMS THIS AERO BOX GROUP LOADS UP
C
C 400 K=1,KK
C NREAM=NBSEQ(NFBEAM,K)
C
C 300 M=1,NINTLD
C NB=STALDS(M,1)
C IF (NB.NE.NREAM) GO TO 300
C
C XB=STALDS(M,4)
C YB=STALDS(M,5)
C ZB=STALDS(M,6)
C LOADCD=STALDS(M,2)
C
C NOW CALC INT.LOADS FOR ALL UNIT AERO LOADS IN THIS AERO BOX GRO
C
C 200 J=1,NBOXES
C XM=RXIN(J)
C YM=RYIN(J)
C ZM=RHZIN(J)
C DX=XM-XB
C DY=YM-YB
C DZ=ZM-ZB
C
C TEST FOR A AERO BOX IB OF THIS BEAM LOAD STA. AND DON T USE
C
C IF(NR.NE.NFBEAM)GO TC 150
C YLOAD=TLAMY(1,NFBEAM)*XM
C 1 + TLAMY(2,NFBEAM)*YM
C 1 + TLAMY(3,NFBEAM)*ZM
C 1 + TLAMY(4,NFBEAM)
C YFEAM=TLAMY(1,NFBEAM)*XB

```

GEOMAB54
 GEOMAB55
 GEOMAB56
 GEOMAB57
 GEOMAB58
 GEOMAB59
 GEOMAB60
 GEOMAB61
 GEOMAB62
 GEOMAB63
 GEOMAB64
 GEOMAB65
 GEOMAB66
 GEOMAB67
 GEOMAB68
 GEOMAB69
 GEOMAB70
 GEOMAB71
 GEOMAB72
 GEOMAB73
 GEOMAB74
 GEOMAB75
 GEOMAB76
 GEOMAB77
 GEOMAB78
 GEOMAB79
 GEOMAB80
 GEOMAB81
 GEOMAB82
 GEOMAB83
 GEOMAB84
 GEOMAB85
 GEOMAB86
 GEOMAB87
 GEOMAB88
 GEOMAB89
 GEOMAB90
 GEOMAB91
 GEOMAB92
 GEOMAB93
 GEOMAB94
 GEOMAB95
 GEOMAB96
 GEOMAB97
 GEOMAB98
 GEOMAB99
 GEOMAB100
 GEOMAB101
 GEOMAB102
 GEOMAB103
 GEOMAB104
 GEOMAB105

```

1 +TLAMY(2,NFBEAM)*YR
1 +TLAMY(3,NFBEAM)*ZB
1 +TLAMY(4,NFBEAM)
1 IF (ABS(YLOAD).LT.ABS(YBEAM))GO TO 200
C 150 CCNTI NUF
C
C 200 CCNTI NUF
300 CCNTI NUF
400 CCNTI NUF
C
      CALL ABAMLT (LOADCD,PINTP(M,L,J),TLAMM(1,1,NB),DX,DY,DZ,GAMZ,GAMY)
C 300 CCNTI NUF
300 CCNTI NUF
400 CCNTI NUF
C
      IF (KPRLDS.EQ.0)GO TO 420
      J1=1
      J2=7
      I1=NFNLRR(1,L)
      I2=I1+6
      IF (J2.GT.NBOXES) J2=NBOXES
      WRITE(NOUT,9)(I,I=I1,I2)
9   FORMAT (1H0,4X,7I14)
      DO 410 M=1,NINTLD
      410 WRITE(NOUT,10)M,(PINTP(M,L,J),J=J1,J2)
10  FORMAT (1H,14,7E14.6)
      IF (J2.GF.NBOXES)GO TO 420
      J1=J2+1
      J2=J1+6
      I1=I2+1
      I2=I1+6
      GO TO 405
420 CCNTI NUF
C 500 CCNTI NUF
C
      RETURN
      END

```

GEOMAI06
 GEOMAI07
 GEOMAI08
 GEOMAI09
 GEOMAI10
 GEOMAI11
 GEOMAI12
 GEOMAI13
 GEOMAI14
 GEOMAI15
 GEOMAI16
 GEOMAI17
 GEOMAI18
 GEOMAI19
 GEOMAI20
 GEOMAI21
 GEOMAI22
 GEOMAI23
 GEOMAI24
 GEOMAI25
 GEOMAI26
 GEOMAI27
 GEOMAI28
 GEOMAI29
 GEOMAI30
 GEOMAI31
 GEOMAI32
 GEOMAI33
 GEOMAI34
 GEOMAI35
 GEOMAI36
 GEOMAI37
 GEOMAI38
 GEOMAI39
 GEOMAI40
 GEOMAI41
 GEOMAI42

```

SUBROUTINE GEOMMB(NMGRP,NFNLMB,NBEAMS,NRCON,NBSEQ,NINTLD,STALDS,
1 PINTF,PINTL,PINTH,PINTI,PINTA,PINTP,
2 ELXIO,FLYIO,ELZIO,FLAMY,FLAMV,FLAMV)
THIS SUBROUTINE CALCULATES THE INTEGRATED LOADS IN TERMS OF ALL
UNIT INERTIAL LOADS AND THEN IN TERMS OF GEN. RESPONSE
DIMENSION NFNLMB(3,1)
DIMENSION NBSEQ(NBEAMS,1)
DIMENSION STALDS(NINTLD,8)
DIMENSION PINTF(NINTLD,NMGRP,1)
DIMENSION PINTL(NINTLD,NMGRP,1)
DIMENSION PINTH(NINTLD,NMGRP,1)
DIMENSION PINTI(NINTLD,NMGRP,1)
DIMENSION PINTA(NINTLD,NMGRP,1)
DIMENSION PINTP(NINTLD,NMGRP,1)
DIMENSION ELXIO(1),ELZIO(1)
DIMENSION FLAMY(4,1)
DIMENSION FLAMV(3,3,1)
DIMENSION FLAMV(3,3,1)
COMMON /Z7Z/HEDR(49),NOUT,KROW,LINES,IPRNT,NER
COMMON/XTF/NF(100)
EQUIVALENCE (NF(24),KPRLDS), (NF(80),ISECT)
PINTH,L,F ARE INT.LDS. AT DESIGNATED STA. DUE TO H,L,F FORCES
WHERE H,L,F ARE EQUILIBRIUM FORCES IN THE +Z,+Y,+X AAS DIRECTIONS
M=INTGD LD. NO.
L=MASS GRP NO.
J=MASS NO. IN LTH GRP.
LOOP THRU ALL MASS GROUPS
DO 500 L=1,NMGRP
NMASS=NFNLMB(2,L)-NFNLMB(1,L)+1
NRFAM=NFNLMB(3,L)
KK=NBSEQ(NFBEAM)
DO 40 M=1,NINTLD
DO 40 J=1,NMASS
PINTH(M,L,J)=0.0
PINTL(M,L,J)=0.0
PINTF(M,L,J)=0.0
PINTI(M,L,J)=0.0
PINTA(M,L,J)=0.0
PINTP(M,L,J)=0.0
40 PINTP(M,L,J)=0.0
WRITE(NOUT,8)L,NFNLMB(1,L),NFNLMB(2,L)
FORMAT(1H1,20X,'INERTIAL LOAD DATA')
1 1H0,20X,'GEOMETRY FOR MASS GROUP',I4,' * MASS',

```

GEOMMB 2
 GEOMMB 3
 GEOMMB 4
 GEOMMB 5
 GEOMMB 6
 GEOMMB 7
 GEOMMB 8
 GEOMMB 9
 GEOMMB 10
 GEOMMB 11
 GEOMMB 12
 GEOMMB 13
 GEOMMB 14
 GEOMMB 15
 GEOMMB 16
 GEOMMB 17
 GEOMMB 18
 GEOMMB 19
 GEOMMB 20
 GEOMMB 21
 GEOMMB 22
 GEOMMB 23
 GEOMMB 24
 GEOMMB 25
 GEOMMB 26
 GEOMMB 27
 GEOMMB 28
 GEOMMB 29
 GEOMMB 30
 GEOMMB 31
 GEOMMB 32
 GEOMMB 33
 GEOMMB 34
 GEOMMB 35
 GEOMMB 36
 GEOMMB 37
 GEOMMB 38
 GEOMMB 39
 GEOMMB 40
 GEOMMB 41
 GEOMMB 42
 GEOMMB 43
 GEOMMB 44
 GEOMMB 45
 GEOMMB 46
 GEOMMB 47
 GEOMMB 48
 GEOMMB 49
 GEOMMB 50
 GEOMMB 51
 GEOMMB 52
 GEOMMB 53

```

C 1 14,* TD*,I4/IH0,*MASS*,6X,*X*,11X,*Y*,11X,*Z*)
C 100 100 J=1,NMASS
C 100 LOC=NFNLMP(1,L)+J-1
C 100 WRT IF (NOUT,7) LOC,FLXIO(LOC),ELYIO(LOC) FLZIO(LOC)
C 100 CONTINUE
C 100 7 FORMAT (1H ,I4,2E14.6)
C 100
C 100 FIND ALL BEAMS THIS MASS GROUP LOADS UP
C 100
C 100 400 K=1,KK
C 100 NBEAM=NRSEQ(NFBEAM,K)
C 100
C 100 FIND ALL INTEGRATED LOAD STA. THIS MASS GROUP LOADS UP
C 100
C 100 300 M=1,NINTLD
C 100 NB=STALDS(M,1)
C 100 IF (NR.NE.NBEAM) GO TO 300
C 100 XB=STALDS(M,4)
C 100 YB=STALDS(M,5)
C 100 ZB=STALDS(M,6)
C 100 LOADCD=STALDS(M,2)
C 100
C 100 NCW CALC INT.LOADS FOR ALL UNIT INERTIA LOADS IN THIS MASS GROUP
C 100
C 100 200 J=1,NMASS
C 100 LOC=NFNLMP(1,L)+J-1
C 100 XM=ELXIO(LOC)
C 100 YM=ELYIO(LOC)
C 100 ZN=FLZIO(LOC)
C 100 X=XM-XB
C 100 Y=YM-YB
C 100 Z=ZN-ZB
C 100
C 100 TEST FOR A MASS IR OF THIS MASS GRP ENTRY BEAM DON T USE IT
C 100
C 100 IF (NB.NE.NFBEAM) GO TO 150
C 100 YLOAD=TLAMY(1,NFBEAM)*XM
C 100 1 + TLAMY(2,NFBEAM)*YM
C 100 1 + TLAMY(3,NFBEAM)*ZN
C 100 1 + TLAMY(4,NFBEAM)*XB
C 100 YBEAM=TLAMY(1,NFBEAM)*XB
C 100 1 + TLAMY(2,NFBEAM)*YB
C 100 1 + TLAMY(3,NFBEAM)*ZB
C 100 1 + TLAMY(4,NFBEAM)*ZB
C 100 IF (ABS(YLOAD).LT.ABS(YBEAM)) GO TO 200
C 100
C 150 CONTINUE
C 100
C 100 LOADCD DESIGNATES INTG D LOAD TYPE

```

GEOMM854
 GEOMM855
 GEOMM856
 GEOMM857
 GEOMM858
 GEOMM859
 GEOMM860
 GEOMM861
 GEOMM862
 GEOMM863
 GEOMM864
 GEOMM865
 GEOMM866
 GEOMM867
 GEOMM868
 GEOMM869
 GEOMM870
 GEOMM871
 GEOMM872
 GEOMM873
 GEOMM874
 GEOMM875
 GEOMM876
 GEOMM877
 GEOMM878
 GEOMM879
 GEOMM880
 GEOMM881
 GEOMM882
 GEOMM883
 GEOMM884
 GEOMM885
 GEOMM886
 GEOMM887
 GEOMM888
 GEOMM889
 GEOMM890
 GEOMM891
 GEOMM892
 GEOMM893
 GEOMM894
 GEOMM895
 GEOMM896
 GEOMM897
 GEOMM898
 GEOMM899
 GEOMM900
 GEOMM901
 GEOMM902
 GEOMM903
 GEOMM904
 GEOMM905
 GEOMM906
 GEOMM907
 GEOMM908
 GEOMM909
 GEOMM910
 GEOMM911
 GEOMM912
 GEOMM913
 GEOMM914
 GEOMM915
 GEOMM916
 GEOMM917
 GEOMM918
 GEOMM919
 GEOMM920
 GEOMM921
 GEOMM922
 GEOMM923
 GEOMM924
 GEOMM925
 GEOMM926
 GEOMM927
 GEOMM928
 GEOMM929
 GEOMM930
 GEOMM931
 GEOMM932
 GEOMM933
 GEOMM934
 GEOMM935
 GEOMM936
 GEOMM937
 GEOMM938
 GEOMM939
 GEOMM940
 GEOMM941
 GEOMM942
 GEOMM943
 GEOMM944
 GEOMM945
 GEOMM946
 GEOMM947
 GEOMM948
 GEOMM949
 GEOMM950
 GEOMM951
 GEOMM952
 GEOMM953
 GEOMM954
 GEOMM955
 GEOMM956
 GEOMM957
 GEOMM958
 GEOMM959
 GEOMM960
 GEOMM961
 GEOMM962
 GEOMM963
 GEOMM964
 GEOMM965
 GEOMM966
 GEOMM967
 GEOMM968
 GEOMM969
 GEOMM970
 GEOMM971
 GEOMM972
 GEOMM973
 GEOMM974
 GEOMM975
 GEOMM976
 GEOMM977
 GEOMM978
 GEOMM979
 GEOMM980
 GEOMM981
 GEOMM982
 GEOMM983
 GEOMM984
 GEOMM985
 GEOMM986
 GEOMM987
 GEOMM988
 GEOMM989
 GEOMM990
 GEOMM991
 GEOMM992
 GEOMM993
 GEOMM994
 GEOMM995
 GEOMM996
 GEOMM997
 GEOMM998
 GEOMM999
 GEOMM1000
 GEOMM1001
 GEOMM1002
 GEOMM1003
 GEOMM1004
 GEOMM1005

```

C 1=M
C 2=T
C 3=Y
C 4=H
C 5=L
C 6=F
      CALL MMBMLT (LOADCD, ISECT, TLAMM(1,1,NB), TLAMV(1,1,L), DX,DY,DZ
1 , PINTF(M,L,J), PINTL(M,L,J), PINTH(M,L,J)
2 , PINTT(M,L,J), PINTA(M,L,J), PINTP(M,L,J) )
C
C 200 CONTINUE
C 300 CONTINUE
C 400 CONTINUE
      IF (KPRLDS.EQ.0) GO TO 490
      WRITE (NOUT,6)
6      FORMAT (1H0,20X,*INTEGRATED LOADS PER INERTIA LOADS*/1H0,
1 20X,*Z LOADS*/1H0,*LOAD*,20X,*MASS NOS.*)
9      FORMAT (1H0,4X,7I14)
10     FORMAT (1H ,14,7E14.6)
11     FORMAT (1H0,20X,*Y LOADS*/1H0,20X,*MASS NOS.*)
12     FORMAT (1H0,20X,*X LOADS*/1H0,20X,*MASS NOS.*)
13     FORMAT (1H0,20X,*Z MOMENTS*/1H0,20X,*MASS NOS.*)
14     FORMAT (1H0,20X,*Y MOMENTS*/1H0,20X,*MASS NOS.*)
15     FORMAT (1H0,20X,*X MOMENTS*/1H0,20X,*MASS NOS.*)
C
C 10 = NFNLMB(1,L) - 1
C
DC 470 J1=1,NMASS,7
J2 = J1+6
IF (J2.GT.NMASS) J2=NMASS
I1 = 10+J1
I2 = 10+J2
WRITE (NOUT,9) (I,I=11,I2)
C 410 M=1,NINTLD
      WRITE (NOUT,10) M, (PINTH(M,L,J),J=J1,J2)
      WRITE (NOUT,11)
      WRITE (NOUT,9) (I,I=11,I2)
C 420 M=1,NINTLD
      WRITE (NOUT,10) M, (PINTL(M,L,J),J=J1,J2)
      WRITE (NOUT,12)
      WRITE (NOUT,9) (I,I=11,I2)
C 430 M=1,NINTLD
      WRITE (NOUT,10) M, (PINTF(M,L,J),J=J1,J2)
      WRITE (NOUT,13)
      WRITE (NOUT,9) (I,I=11,I2)
C 440 M=1,NINTLD
      WRITE (NOUT,10) M, (PINTP(M,L,J),J=J1,J2)
      WRITE (NOUT,14)
      WRITE (NOUT,9) (I,I=11,I2)
C 450 M=1,NINTLD

```

GEOMM106
 GEOMM107
 GEOMM108
 GEOMM109
 GEOMM110
 GEOMM111
 GEOMM112
 GEOMM113
 GEOMM114
 GEOMM115
 GEOMM116
 GEOMM117
 GEOMM118
 GEOMM119
 GEOMM120
 GEOMM121
 GEOMM122
 GEOMM123
 GEOMM124
 GEOMM125
 GEOMM126
 GEOMM127
 GEOMM128
 GEOMM129
 GEOMM130
 GEOMM131
 GEOMM132
 GEOMM133
 GEOMM134
 GEOMM135
 GEOMM136
 GEOMM137
 GEOMM138
 GEOMM139
 GEOMM140
 GEOMM141
 GEOMM142
 GEOMM143
 GEOMM144
 GEOMM145
 GEOMM146
 GEOMM147
 GEOMM148
 GEOMM149
 GEOMM150
 GEOMM151
 GEOMM152
 GEOMM153
 GEOMM154
 GEOMM155
 GEOMM156
 GEOMM157

```

450 WRITE (NCUT,10) M, (PINTA(M,L,J),J=J1,J2)
    WRITE (NCUT,15)
    WRITE (NCUT,9) (I,I=11,12)
    DO 460 M=1,NINTLD
460  WRITE (NCUT,10) M, (PINTI(M,L,J),J=J1,J2)
      C 470 CONTINUE
      C 490 CONTINUE
      C 500 CONTINUE
      C
      RETURN
      END

```

```

GEOMMI58
GEOMMI59
GEOMMI60
GEOMMI61
GEOMMI62
GEOMMI63
GEOMMI64
GEOMMI65
GEOMMI66
GEOMMI67
GEOMMI68
GEOMMI69
GEOMMI70
GEOMMI71
GEOMMI72

```

```

SUBROUTINE GEOMSB(NSBGRP,NFNLSB,NREAMS,NBCON,NRSEQ,NINTLD,STALDS,
1 PINTZ,PINTY,SBXIO,SBYIO,SBZIO,FLAMY,FLAMM,
1 XIS1,ETAS1,ZETAS1,XIS2,ETAS2,ZETAS2)
THIS SUBROUTINE CALCS THE INTEGRATED LOADS IN TERMS OF ALL
UNIT S.B. AERO LOADS AND THEN IN TERMS OF GFN. RESPONSE
DIMENSION NFNLSB(3,1)
DIMENSION NBCON(1)
DIMENSION NBSFO(NBEAMS,1)
DIMENSION STALDS(NINTLD,8)
DIMENSION PINTZ(NINTLD,NSBGRP,1)
DIMENSION PINTY(NINTLD,NSBGRP,1)
DIMENSION XIS1(1),XIS2(1)
DIMENSION ETAS1(1),ETAS2(1)
DIMENSION ZETAS1(1),ZETAS2(1)
DIMENSION SBXIO(1),SBYIO(1),SBZIO(1)
DIMENSION FLAMY(4,1)
DIMENSION FLAMM(3,3,1)
COMMON /ZZZ/HEDF(49),NOUT,KROW,LINES,IPRNT,NFR
COMMON/XTE/NF(100)
EQUIVALENCE (NF(24),KPRLDS)
LOOP THRU ALL S.B. GROUPS
DO 500 L=1,NSBGRP
NAERSR=NFNLSB(2,L)-NFNLSB(1,L)+1
NFBREAM=NFNLSB(3,L)
KK=NBCON(NFBREAM)
DO 40 M=1,NINTLD
DO 40 J=1,NAERSR
PINTY(M,L,J) = 0.0
40 PINTZ(M,L,J) = 0.0
WRITE (NOUT,8)L,NFNLSB(1,L),NFNLSB(2,L)
FORMAT(1H1,20X,*BODY LOAD DATA*/1H0,20X,
1 *GEOMETRY FOR BODY GROUP*,14,* BODIES*,
1 14,* TO*,14/1H0,*BODY*,6X,*X*,11X,*Y*,11X,*Z*)
50 CONTINUE
FIND DIR CGS AND S.B. FORCE COORD
LOC=NFNLSB(1,L)
GAMY=1.0
IF (XIS2(LOC).LT.XIS1(LOC))GAMY=-1.0
GAMZ=1.0
DO 100 J=1,NAERSR
LOC=NFNLSB(1,L)+J-1
SBXIO(J)=(XIS2(LOC)+XIS1(LOC))/2.
SBYIO(J)=(ETAS2(LOC)+ETAS1(LOC))/2.

```

GEOMSB 2
 GEOMSB 3
 GEOMSB 4
 GEOMSB 5
 GEOMSB 6
 GEOMSB 7
 GEOMSB 8
 GEOMSB 9
 GEOMSB 10
 GEOMSB 11
 GEOMSB 12
 GEOMSB 13
 GEOMSB 14
 GEOMSB 15
 GEOMSB 16
 GEOMSB 17
 GEOMSB 18
 GEOMSB 19
 GEOMSB 20
 GEOMSB 21
 GEOMSB 22
 GEOMSB 23
 GEOMSB 24
 GEOMSB 25
 GEOMSB 26
 GEOMSB 27
 GEOMSB 28
 GEOMSB 29
 GEOMSB 30
 GEOMSB 31
 GEOMSB 32
 GEOMSB 33
 GEOMSB 34
 GEOMSB 35
 GEOMSB 36
 GEOMSB 37
 GEOMSB 38
 GEOMSB 39
 GEOMSB 40
 GEOMSB 41
 GEOMSB 42
 GEOMSB 43
 GEOMSB 44
 GEOMSB 45
 GEOMSB 46
 GEOMSB 47
 GEOMSB 48
 GEOMSB 49
 GEOMSB 50
 GEOMSB 51
 GEOMSB 52
 GEOMSB 53

```

      SBZIN(J)=(ZETAS2(LCC)+7*ETAS1(LCC))/2.
      WRITE(NCUT,7) LCC,SBXIN(J),SBYIN(J),SBZIN(J)
7  FORMAT (1H,14,3F14.6)
C 100 CONTINUE
      IF(KPRLOS.FQ.J)GO TO 105
      WRITE(NCUT,6)
      FORMAT(1H0,20X,'INTEGRATED LOAD PER Z BODY LOAD#/'
1 140,'LOAD#',20X,'BODY NOS.#')
C 105 CONTINUE
      FIND ALL BEAMS THIS S.B. GROUP LOADS UP
      DO 400 K=1,KK
      NPEAM=NPSEQ(NFBEAM,K)
      FIND ALL INTEGRATED LOAD STA. THIS S.B. GROUP LOADS UP
      DO 300 M=1,NINTLD
      NP=STALDS(M,1)
      IF (NB.NF.NBEAM) GO TO 300
      XP=STALDS(M,4)
      YB=STALDS(M,5)
      ZF=STALDS(M,6)
      LOADCD=STALDS(M,2)
      NOW CALC INT.LOADS FOR ALL UNIT S.B.AERO LOADS IN THIS S.B. GROUP
      DO 200 J=1,NAFRSR
      XM=SBXIN(J)
      YM=SBYIN(J)
      ZA=SBZIN(J)
      DX=XM-XR
      DY=YM-YR
      DZ=ZM-ZR
      TEST FOR A S.B. IP OF THIS S.B. GRP ENTRY BFAM DON T USE IT
      IF (NB.NF.NFBEAM)GO TO 150
      VLQAD=TLAMY(1,NFBEAM)*XM
      1 + TLAMY(2,NFBEAM)*YM
      1 + TLAMY(3,NFBEAM)*ZM
      1 + TLAMY(4,NFBEAM)
      VPFAM=TLAMY(1,NFBEAM)*XB
      1 + TLAMY(2,NFBEAM)*YR
      1 + TLAMY(3,NFBEAM)*ZR
      1 + TLAMY(4,NFBEAM)

```

GEOMSB54
 GEOMSB55
 GEOMSB56
 GEOMSB57
 GEOMSB58
 GEOMSB59
 GEOMSB60
 GEOMSB61
 GEOMSB62
 GEOMSB63
 GEOMSB64
 GEOMSB65
 GEOMSB66
 GEOMSB67
 GEOMSB68
 GEOMSB69
 GEOMSB70
 GEOMSB71
 GEOMSB72
 GEOMSB73
 GEOMSB74
 GEOMSB75
 GEOMSB76
 GEOMSB77
 GEOMSB78
 GEOMSB79
 GEOMSB80
 GEOMSB81
 GEOMSB82
 GEOMSB83
 GEOMSB84
 GEOMSB85
 GEOMSB86
 GEOMSB87
 GEOMSB88
 GEOMSB89
 GEOMSB90
 GEOMSB91
 GEOMSB92
 GEOMSB93
 GEOMSB94
 GEOMSB95
 GEOMSB96
 GEOMSB97
 GEOMSB98
 GEOMSB99
 GEOMSI00
 GEOMSI01
 GEOMSI02
 GEOMSI03
 GEOMSI04
 GEOMSI05

GEOMSI06
GEOMSI07
GEOMSI08
GEOMSI09
GEOMSI10
GEOMSI11
GEOMSI12
GEOMSI13
GEOMSI14
GEOMSI15
GEOMSI16
GEOMSI17
GEOMSI18
GEOMSI19
GEOMSI20
GEOMSI21
GEOMSI22
GEOMSI23
GEOMSI24
GEOMSI25
GEOMSI26
GEOMSI27
GEOMSI28
GEOMSI29
GEOMSI30
GEOMSI31
GEOMSI32
GEOMSI33
GEOMSI34
GEOMSI35
GEOMSI36
GEOMSI37
GEOMSI38
GEOMSI39
GEOMSI40
GEOMSI41
GEOMSI42
GEOMSI43
GEOMSI44
GEOMSI45
GEOMSI46
GEOMSI47
GEOMSI48
GEOMSI49
GEOMSI50
GEOMSI51
GEOMSI52
GEOMSI53
GEOMSI54
GEOMSI55
GEOMSI56
GEOMSI57

```

C 150 CONTINUE
      CALL SBAMLT (LOADCD,PINIZ(M,L,J),PINTY(M,L,J),TLAMM(1,1,NB)
1,DX,DY,DZ,GAMZ,GAMY)
C 200 CONTINUE
C 300 CONTINUE
C 400 CONTINUE
      IF (KPRLOS.EQ.0) GO TO 440
      J1=1
      J2=7
      I1=NF NLSR(1,L)
      I2=I1+6
      IF (J2.GT.NAERSB) J2=NAERSB
      WRITE (NOUT,9) (I,I=I1,I2)
9      FORMAT (1H,4X,7I14)
      DO 410 M=1,NINTLD
410    WRITE (NOUT,10) M,(PINTZ(M,L,J),J=J1,J2)
10    FORMAT (1H,14,7E14.6)
      IF (J2.GE.NAERSB) GO TO 420
      J1=J2+1
      J2=J1+6
      I1=I2+1
      I2=I1+6
      GO TO 405
420    CONTINUE
      WRITE (NOUT,5)
5      FORMAT (1H0,20X,*INTEGRATED LOADS PER Y BODY LOAD*/
1,140,*LOAD*,20X,*BODY NOS.*)
      J1=1
      J2=7
      I1=NF NLSR(1,L)
      I2=I1+6
      IF (J2.GT.NAERSB) J2=NAERSB
      WRITE (NOUT,9) (I,I=I1,I2)
9      FORMAT (1H,4X,7I14)
      DO 430 M=1,NINTLD
430    WRITE (NOUT,10) M,(PINTY(M,L,J),J=J1,J2)
10    FORMAT (1H,14,7E14.6)
      IF (J2.GE.NAERSB) GO TO 440
      J1=J2+1
      J2=J1+6
      I1=I2+1
      I2=I1+6
      GO TO 425
C 440 CONTINUE
C 500 CONTINUE
C      RETURN
      END

```

```

SUBROUTINE INPUTL (NTI, BFAMGM, NBEAMS, STALDS, NINTLD, STRESS, NSTRSS
1 , RFNLMB, NMGRP, RFNLAB, NABGRP, RFNLB, NSHGRP, NMASS, NBOXES, NAERSB
2 , TLAMV, NENG, ENGM)
COMMON/ZZZ/HEDR(48), NTN, NOUT, KROW, LINES, IPRNT, NER
C
C
DIMENSION BFAMGM(NRFAMS,1), STALDS(NINTLD,1), STRESS(NSTRSS,1)
DIMENSION RFNLMB(3,1), RFNLAB(3,1), RFNLB(3,1)
DIMENSION TLAMV(3,3,1)
DIMENSION ENGM(1)
C
310 FORMAT (6F12.0)
C
DO 100 I=1,NBEAMS
  READ (NTI,310) (BFAMGM(I,J),J=1,3)
  READ (NTI,310) (BEAMGM(I,J),J=4,7)
100 CONTINUE
CALL HEADNG
WRITE (NOUT,1)
FORMAT(1H0,20X, UNIT LOAD MODULE ANALYSIS )
CALL PRNT (BEAMGM,NBEAMS,7,NBEAMS,1,6HBEAMGM,2)
C
DO 120 I=1,NINTLD
  READ (NTI,310) (STALDS(I,J),J=1,3)
  READ (NTI,310) (STALDS(I,J),J=4,8)
120 CONTINUE
CALL PRNT (STALDS,NINTLD,8,NINTLD,1,6HSTALDS,2)
C
IF (NSTRSS.EQ.0) GO TO 150
DO 140 I=1,NSTRSS
  READ (NTI,310) (STRESS(I,J),J=1,NINTLD)
140 CONTINUE
CALL PRNT (STRESS,NSTRSS,NINTLD,NSTRSS,1,6HSTRESS,2)
C
NMASS = 0
DO 200 J=1,NMGRP
  READ (NTI,310) (RFNLMB(I,J),I=1,3)
  ITEST = RFNLMB(2,J) - RFNLMB(1,J) + 1
  NMASS = MAX0(NMASS,ITEST)
200 CONTINUE
CALL PRNT (RFNLMB,3,NMGRP,3,1,6HNFNLMB,2)
C
DO 205 N=1,NMGRP
  DO 204 I=1,3
    READ (NTI,310) (TLAMV(I,J,N),J=1,3)
  WRITE (NOUT,202) N
  FORMAT(1H0, MASS GROUP = ,I4)
  205 CALL PRNT (TLAMV(1,1,N),3,3,3,1,4H1AMV,1)
C
NAERSB = 0
DO 210 J=1,NMGRP
  READ (NTI,310) (RFNLAB(I,J),I=1,3)

```

```

INPUTL 2
INPUTL 3
INPUTL 4
INPUTL 5
INPUTL 6
INPUTL 7
INPUTL 8
INPUTL 9
INPUTL 10
INPUTL 11
INPUTL 12
INPUTL 13
INPUTL 14
INPUTL 15
INPUTL 16
INPUTL 17
INPUTL 18
INPUTL 19
INPUTL 20
INPUTL 21
INPUTL 22
INPUTL 23
INPUTL 24
INPUTL 25
INPUTL 26
INPUTL 27
INPUTL 28
INPUTL 29
INPUTL 30
INPUTL 31
INPUTL 32
INPUTL 33
INPUTL 34
INPUTL 35
INPUTL 36
INPUTL 37
INPUTL 38
INPUTL 39
INPUTL 40
INPUTL 41
INPUTL 42
INPUTL 43
INPUTL 44
INPUTL 45
INPUTL 46
INPUTL 47
INPUTL 48
INPUTL 49
INPUTL 50
INPUTL 51
INPUTL 52
INPUTL 53

```

INPUTL54
 INPUTL55
 INPUTL56
 INPUTL57
 INPUTL58
 INPUTL59
 INPUTL60
 INPUTL61
 INPUTL62
 INPUTL63
 INPUTL64
 INPUTL65
 INPUTL66
 INPUTL67
 INPUTL68
 INPUTL69
 INPUTL70
 INPUTL71
 INPUTL72
 INPUTL73
 INPUTL74
 INPUTL75
 INPUTL76
 INPUTL77
 INPUTL78

```

    ITEST = RFNLAR(2,J) - RFNLAR(1,J) + 1
    NBOXES = MAXO(NBOXES, ITEST)
210 CONTINUE
    CALL PRNT (RFNLAR,3,NABGRP,3,1,6HNFNLAR,2)

    IF (NSBGRP.EQ.0) GO TO 250
    NAERSR = 0
    DO 220 J=1,NSBGRP
      READ (NTI,310) (RFNLSR(I,J),I=1,3)
      ITEST = RFNLSR(2,J) - RFNLSR(1,J) + 1
      NAERSR = MAXO(NAERSR, ITEST)
220 CONTINUE
    CALL PRNT (RFNLSR,3,NSBGRP,3,1,6HNFNLSR,2)

    IF (NENGS.EQ.0) GO TO 260
    II=-1
    DO 255 J=1,NENGS
      II=II+2
      READ (NTI,310) ENGGM(II), ENGGM(II+1)
255 CONTINUE
    CALL PRNT (ENGGM,NENGS,2,NENGS,1,4HENGGM,1)
260 CONTINUE

    RETURN
  END
  
```


MMBMLT154
MMBMLT155
MMBMLT156
MMBMLT157
MMBMLT158
MMBMLT159
MMBMLT160
MMBMLT161
MMBMLT162
MMBMLT163
MMBMLT164
MMBMLT165
MMBMLT166
MMBMLT167
MMBMLT168
MMBMLT169
MMBMLT170

```

C      NOW ROTATE TO INERTIAL LOAD COORDS
      PINTF=AF*TLAMV(1,1)+AL*TLAMV(2,1)+AH*TLAMV(3,1)
      PINTL=AF*TLAMV(1,2)+AL*TLAMV(2,2)+AH*TLAMV(3,2)
      PINTH=AF*TLAMV(1,3)+AL*TLAMV(2,3)+AH*TLAMV(3,3)
      IF (I SECT.EQ.0) GO TO 600
      PINTT = 0.0
      PINTA = 0.0
      PINTP = 0.0
      IF (LOADCD.GT.3) GO TO 600
      PINTT = AT*TLAMV(1,1) + AA*TLAMV(2,1) + AP*TLAMV(3,1)
      PINTA = AT*TLAMV(1,2) + AA*TLAMV(2,2) + AP*TLAMV(3,2)
      PINTP = AT*TLAMV(1,3) + AA*TLAMV(2,3) + AP*TLAMV(3,3)
C      600 RETURN
      END

```

MSSPHI 2
 MSSPHI 3
 MSSPHI 4
 MSSPHI 5
 MSSPHI 6
 MSSPHI 7
 MSSPHI 8
 MSSPHI 9
 MSSPHI 10
 MSSPHI 11
 MSSPHI 12
 MSSPHI 13
 MSSPHI 14
 MSSPHI 15
 MSSPHI 16
 MSSPHI 17
 MSSPHI 18
 MSSPHI 19
 MSSPHI 20
 MSSPHI 21
 MSSPHI 22
 MSSPHI 23
 MSSPHI 24
 MSSPHI 25
 MSSPHI 26
 MSSPHI 27
 MSSPHI 28
 MSSPHI 29
 MSSPHI 30
 MSSPHI 31
 MSSPHI 32
 MSSPHI 33
 MSSPHI 34
 MSSPHI 35
 MSSPHI 36
 MSSPHI 37
 MSSPHI 38
 MSSPHI 39
 MSSPHI 40
 MSSPHI 41
 MSSPHI 42
 MSSPHI 43
 MSSPHI 44
 MSSPHI 45
 MSSPHI 46
 MSSPHI 47
 MSSPHI 48
 MSSPHI 49
 MSSPHI 50
 MSSPHI 51
 MSSPHI 52
 MSSPHI 53

SUBROUTINE MSSPHI (NMGRP,NFNLMB,NINTLD,
 1 , PINTF, PINTL, PINTH, PINTT, PINTA, PINTP
 2 , NDOF,NMS,FMPHI,NM,PIQ,SYMCOD,ASMCOD,NSYM,NASYM)

COMMON/XTF/NF(100)
 EQUIVALENCE (NF(24),KPRCHK)

DIMENSION PIQ(NINTLD,1)
 DIMENSION PINTF(NINTLD,NMGRP,1)
 DIMENSION PINTL(NINTLD,NMGRP,1)
 DIMENSION PINTH(NINTLD,NMGRP,1)
 DIMENSION PINTT(NINTLD,NMGRP,1)
 DIMENSION PINTA(NINTLD,NMGRP,1)
 DIMENSION PINTP(NINTLD,NMGRP,1)
 DIMENSION FMPHI(NDOF,NMS,1)
 DIMENSION NFNLMB(3,1)
 DIMENSION SYMCOD(1),ASMCOD(1)

CALC INTGD INERTIAL LOADS DUE TO UNIT MODAL AMPLITUDES

DO 100 M=1,NINTLD
 DO 100 N=1,NM
 PIQ(M,N)=0.
 DC 100 L=1,NMGRP
 NPASS=NFNLMB(2,L)-NFNLMB(1,L)+1
 DO 90 J=1,NPASS
 LCC=NFNLMB(1,L)+J-1
 PIQ(M,N)=PIQ(M,N)+PINTF(M,L,J)*FMPHI(1,LCC,N)
 1 + PINTL(M,L,J)*FMPHI(2,LCC,N)
 1 + PINTH(M,L,J)*FMPHI(3,LCC,N)
 IF (NDOF.LT.4) GO TO 90
 PIQ(M,N) = PIQ(M,N) + PINTT(M,L,J) * FMPHI(4,LCC,N)
 IF (NDOF.LT.5) GO TO 90
 PIQ(M,N) = PIQ(M,N) + PINTA(M,L,J) * FMPHI(5,LCC,N)
 IF (NDOF.LT.6) GO TO 90
 PIQ(M,N) = PIQ(M,N) + PINTP(M,L,J) * FMPHI(6,LCC,N)
 90 CONTINUE
 100 CONTINUE

NOW CHECK FOR SYM,ASYM CORRECTIONS ON C/L LOADS

DC 300 M=1,NINTLD
 IF (SYMCOD(M).EQ.1.AND.ASMCOD(M).EQ.1)GO TO 300
 DO 200 N=1,NSYM
 200 PIQ(M,N)=PIQ(M,N)*SYMCOD(M)
 DC 250 N=1,NASYM
 NN=N+NSYM
 250 PIQ(M,NN)=PIQ(M,N)*ASMCOD(M)
 300 CONTINUE

MSSPHI54
MSSPHI55
MSSPHI56
MSSPHI57
MSSPHI58

IF (KPRC-K. NF.0)CALL PRNT(PIQ,NINTLD,NM,NINTLD,1,4H PIQ ,1)

RETURN
END

C
C

2 3 4 5 6 7
 RTOI
 RTOI
 RTOI
 RTOI
 RTOI

SUBROUTINE RTCI (R,I,N)
 DIMENSION R(1),I(1)
 DO 100 M=1,N
 100 I(M) = R(M)
 RETURN
 END

```

C          SUBROUTINE SBAMLT(LOADCD,PINTZ,PINTY,TLAMM,DX,DY,DZ,GAMZ,GAMY)
C          DIMENSION TLAMM(3,3)
C          GO TO (501,502,503,504,505,506),LOADCD
C
C          501 AL=-TLAMM(1,1)*DZ+TLAMM(1,3)*DX
C              AH=TLAMM(1,1)*DY-TLAMM(1,2)*DX
C              GO TO 510
C
C          502 AL=-TLAMM(2,1)*DZ+TLAMM(2,3)*DX
C              AH=TLAMM(2,1)*DY-TLAMM(2,2)*DX
C              GO TO 510
C
C          503 AL=-TLAMM(3,1)*DZ+TLAMM(3,3)*DX
C              AH=TLAMM(3,1)*DY-TLAMM(3,2)*DX
C              GO TO 510
C
C          504 AL=TLAMM(3,2)
C              AH=TLAMM(3,3)
C              GO TO 510
C
C          505 AL=TLAMM(2,2)
C              AH=TLAMM(2,3)
C              GO TO 510
C
C          506 AL=TLAMM(1,2)
C              AH=TLAMM(1,3)
C
C          510 CONTINUE
C          NEW ROTATE TO AERO LOAD SYSTEM
C          PINTZ=AH*GAMZ
C          PINTY=AL*GAMY
C
C          RETURN
C          END

```

```

SBAMLT 2
SBAMLT 3
SBAMLT 4
SBAMLT 5
SBAMLT 6
SBAMLT 7
SBAMLT 8
SBAMLT 9
SBAMLT 10
SBAMLT 11
SBAMLT 12
SBAMLT 13
SBAMLT 14
SBAMLT 15
SBAMLT 16
SBAMLT 17
SBAMLT 18
SBAMLT 19
SBAMLT 20
SBAMLT 21
SBAMLT 22
SBAMLT 23
SBAMLT 24
SBAMLT 25
SBAMLT 26
SBAMLT 27
SBAMLT 28
SBAMLT 29
SBAMLT 30
SBAMLT 31
SBAMLT 32
SBAMLT 33
SBAMLT 34
SBAMLT 35
SBAMLT 36
SBAMLT 37
SBAMLT 38
SBAMLT 39
SBAMLT 40
SBAMLT 41

```

```

SUBROUTINE SYMSOL(NINTLD,SYMCOD,ASMCOD,STALDS,TLAMM)
C
C
DIMENSION STALDS(NINTLD,1),TLAMM(3,3,1),SYMCOD(1),ASMCOD(1)
C
COMMON/XTF/XF(100)
EQUIVALENCE (XF(24),KPRLDS)
C
DATA EPL/0.01/
C
NOW FIND INTEG LOADS ON C/L, SET UP MATRIX MOD FOR EACH
CHECK FOR C/L LOADS,IF NOT ON C/L THEN IT IS NOT MODED
C
DO 500 I=1,NINTLD
SYMCOD(I)=1.0
ASMCOD(I)=1.0
Y=STALDS(I,5)
IF (ABS(Y).GT.EPL)GO TO 500
C
FOUND C/L LOAD FIND DIR COS, IF SYM LOAD SET ASMCOD=0,
IF ASYM LOAD SET SYMCOD=0, IF MIXED SET APPROP. VAL IN ARRAYS
C
N=STALDS(I,1)
LOADCD=STALDS(I,2)
IF (LOADCD.EQ.1).CR.LOADCD.EQ.6)L=1
IF (LOADCD.EQ.2).CR.LOADCD.EQ.5)L=2
IF (LOADCD).EQ.3).CR.LOADCD.EQ.4)L=3
C
XR=TLAMM(L,1,N)
YR=TLAMM(L,2,N)
ZR=TLAMM(L,3,N)
C
IF (LOADCD.GT.3)GO TO 200
MOMENT CHECK
C
ASYM MOMENT ONLY
IF (ABS(YR).LT.EPL)GO TO 410
SYM MOMENT ONLY
IF (ABS(ZR).LT.EPL.AND.ABS(XR).LT.EPL)GO TO 400
MIXED MOMENT, GET SYM AND ASYM COMPONENTS
C
SYMCOD(I)=ABS(YR)
ASMCOD(I)=SQRT(1.-YR**2)
GO TO 500
C
200 CONTINUE
SHEAR CHECK
C
SYM LOAD ONLY
IF (ABS(YR).LT.EPL)GO TO 400
ASYM LOAD ONLY
IF (ABS(XR).LT.EPL.AND.ABS(ZR).LT.EPL)GO TO 410
C

```

SYMSOL54
 SYMSOL55
 SYMSOL56
 SYMSOL57
 SYMSOL58
 SYMSOL59
 SYMSOL60
 SYMSOL61
 SYMSOL62
 SYMSOL63
 SYMSOL64
 SYMSOL65
 SYMSOL66
 SYMSOL67
 SYMSOL68
 SYMSOL69
 SYMSOL70
 SYMSOL71
 SYMSOL72
 SYMSOL73
 SYMSOL74
 SYMSOL75
 SYMSOL76

```

C
C      MIXED LOAD GET SYM AND ASYM COMPONENTS
C      ASYCOD(I)=ABS(YR)
C      SYMCD(I)=SQRT(1.-Y**2)
C      GO TO 500
C
C      400 CONTINUE
C      ASYCOD(I)=0.
C      GO TO 500
C      410 CONTINUE
C      SYMCD(I)=0.0
C
C      500 CONTINUE
C
C      IF (KPRLOS.EQ.0) GO TO 600
C
C      CALL HEADAG
C      CALL PRNT(SYCOD,NINTLD,1,NINTLD,1,8H SYCOD ,2)
C      CALL PRNT(ASYCOD,NINTLD,1,NINTLD,1,8H ASYCOD ,2)
C      CONTINUE
C      600
C
C      RETURN
C      END
  
```

```

SUBROUTINE THRUST(
1  NENG, NMS, NMGRP, NINTLD, NSYM,
1  ENG, ELXIO, ELYIO, ELZIO, PHIX, PHIY, PHIZ,
1  PINTF, PINTL, PINTH, NFNLMB, TLAMV,
1  THRLOD, THRGNF)
C
COMMON/ZZZ/CASE(48), NIN, NOUT, KPOW, LINES, IPRNT, NER
C
COMMON/XTF/NF(100)
EQUIVALENCE (NF(24), KPRLOS)
C
FORM INTEGRATED LOADS DUE TO UNIT THRUST (THRLOD)
FORM GENLZO FORCE DUE TO UNIT THRUST SYM ONLY (THRGNF)
THRUST ACTS AT FWD MASS OF TWO DEFINED FOR DIRECTION
C
DIMENSION ENG(1), ELXIO(1), ELYIO(1), ELZIO(1)
DIMENSION PHIX(NMS,1), PHIY(NMS,1), PHIZ(NMS,1)
DIMENSION PINTF(NINTLD, NMGRP, 1), PINTL(NINTLD, NMGRP, 1),
1  PINTH(NINTLD, NMGRP, 1)
DIMENSION NFNLMB(3,1), TLAMV(3,3,1)
DIMENSION THRLOD(NINTLD, NENG), THRGNF(NSYM, NENG)
C
C
1  FORMAT(1H0, 'OH OH THIS THRUST MASS NOT DEFINED IN NFNLMB,MSNO ',14)
2  FORMAT(1H0, 'OH OH SOMETHING SCREWED UP, LL= ',14, 'JJ= ',14, 'MS ',14)
3  FORMAT(1H0, '20X, INTEGRATED LOADS DUE TO ENGINE NO. ',14)
4  FORMAT(1H0, '20X, GENLZO FORCE DUE TO ENGINE NO. ',14)
5  FORMAT(20X, 'THRUST LOADS FROM UNIT THRUST')
6  FORMAT(1H0, '20X, ENGINE THRUST MATRICES GENERATED FOR ',14, 'ENG ')
C
C
IF (KPRLOS.NE.0) CALL HEADNG
WRITE (NOUT,6) NENG
IF (KPRLOS.NE.0) WRITE (NOUT,5)
IF NG=0
DO 500 I=1, NENG, 2
C
C
IENG=IENG+1
C
DO 100 K=1, NINTLD
100 THRLOD(K,IENG)=0.
DO 110 K=1, NSYM
110 THRGNF(K,IENG)=0.
C
NFM=ENG M(1)
NLM=ENG M(1+1)
C
C
NC# FIND DIR CCS OF THRUST
C
X1=ELXIO(NFM)
X2=ELYIO(NLM)
Y1=ELYIO(NFM)

```

```

THRUST 2
THRUST 3
THRUST 4
THRUST 5
THRUST 6
THRUST 7
THRUST 8
THRUST 9
THRUST 10
THRUST 11
THRUST 12
THRUST 13
THRUST 14
THRUST 15
THRUST 16
THRUST 17
THRUST 18
THRUST 19
THRUST 20
THRUST 21
THRUST 22
THRUST 23
THRUST 24
THRUST 25
THRUST 26
THRUST 27
THRUST 28
THRUST 29
THRUST 30
THRUST 31
THRUST 32
THRUST 33
THRUST 34
THRUST 35
THRUST 36
THRUST 37
THRUST 38
THRUST 39
THRUST 40
THRUST 41
THRUST 42
THRUST 43
THRUST 44
THRUST 45
THRUST 46
THRUST 47
THRUST 48
THRUST 49
THRUST 50
THRUST 51
THRUST 52
THRUST 53

```


THRUS106
 THRUS107
 THRUS108
 THRUS109
 THRUS110
 THRUS111
 THRUS112
 THRUS113
 THRUS114
 THRUS115
 THRUS116

```

C      IF (KPRLDS.EQ.0)GO TO 500
      WRITE (NOUT,3) IENG
      CALL PRNT(THRLOD(1, IENG), NINTLD, 1, NINTLD, 1, 4H      , 1)
      WRITE (NOUT,4) IENG
      CALL PRNT(THRGNF(1, IENG), NSYM, 1, NSYM, 1, 4H      , 1)
C      500 CONTINUE
C      RETURN
      END

```



```

SUBROUTINE ACSMOD(NTI,
1 NTFM,NTFA,NSYM,NASYM,NOMOD,MXORD,MXBLK,MXOBLK,NMS,
1 PHI,PHISS,PHISA,TMSS,TMSA,TFC,S,TFCA,
1 ANUM,ADEN,ATEN,ATFD,ANUMT,ADENT)
C
C COMMON/ZZZ/CASF(48),NIN,NDUT,KROW,LINES,IPRNT,NER
C
C COMMON/XTF/NF(100)
EQUIVALENCE (NF(81),NDOF)
EQUIVALENCE (NF(71),MXORSN), (NF(72),MXORSO)
1 , (NF(73),MXORAN), (NF(74),MXORAD)
C
DIMENSION TMSS(NTFS,4),TFC(2,NTFS,MXORD),PHISS(NTFS,NSYM)
DIMENSION TMFA(NTFA,4),TFCA(2,NTFA,MXORD),PHISA(NTFA,NASYM)
DIMENSION PHI(1)
DIMENSION ANUM(NTFM,MXBLK,MXOBLK),ADEN(NTFM,MXBLK,MXOBLK)
DIMENSION ATFN(2,MXORD),ATFD(2,MXORD)
DIMENSION ANUMT(MXBLK,MXOBLK),ADENT(MXBLK,MXOBLK)
C
FORMAT(IH3,20X, SYMMETRIC TRANSFER FUNCTION DATA )
FORMAT(IH3,20X, ANTISYMMETRIC TRANSFER FUNCTION DATA )
FORMAT(4F12.0)
C
CALL HEADNG
WRITE(NDUT,1)
C
DO 100 I=1,NTFS
READ(NTI,20) (TMSS(I,J),J=1,4)
C
CALL AFCS (NTFS,TMSS,PHISS,PHI,NSYM,NDOF,NMS,NOMOD,0)
C
CALL TFBLIN(NTI,NTFS,MXBLK,MXOBLK,ANUM,ADEN,0,1.0,1)
C
NOTE TF BLOCKS ARE READ IN TFBLIN
C
CALL TFBLCK(NTFS,MXBLK,MXOBLK,TFC,S,ANUM,ADEN,ATEN,ATFD,
1 ANUMT,ADENT,MXORSN,MXORSO,MXORD)
C
CALL TFEVAL(NTFS,TFC,S,1.0,MXOPSN,MXORSO)
C
C
CALL HEADNG
WRITE(NDUT,2)
C
DO 200 I=1,NTFA
READ(NTI,20) (TMSA(I,J),J=1,4)
C
CALL AFCS (NTFA,TMSA,PHISA,PHI,NASYM,NDOF,NMS,NOMOD,NSYM)
C
CALL TFBLIN(NTI,NTFA,MXBLK,MXOBLK,ANUM,ADEN,0,1.0,1)
C
NOTE TF BLOCKS ARE READ IN TFBLIN
C

```

ACSMOD 2
ACSMOD 3
ACSMOD 4
ACSMOD 5
ACSMOD 6
ACSMOD 7
ACSMOD 8
ACSMOD 9
ACSMOD10
ACSMOD11
ACSMOD12
ACSMOD13
ACSMOD14
ACSMOD15
ACSMOD16
ACSMOD17
ACSMOD18
ACSMOD19
ACSMOD20
ACSMOD21
ACSMOD22
ACSMOD23
ACSMOD24
ACSMOD25
ACSMOD26
ACSMOD27
ACSMOD28
ACSMOD29
ACSMOD30
ACSMOD31
ACSMOD32
ACSMOD33
ACSMOD34
ACSMOD35
ACSMOD36
ACSMOD37
ACSMOD38
ACSMOD39
ACSMOD40
ACSMOD41
ACSMOD42
ACSMOD43
ACSMOD44
ACSMOD45
ACSMOD46
ACSMOD47
ACSMOD48
ACSMOD49
ACSMOD50
ACSMOD51
ACSMOD52
ACSMOD53

```

CALL TFBLOK(NTFA,MXBLK,MXOBLK,TFCA,ANUM,ADEN,ATFN,ATFD,
1 ANUMT,ADENT,MXORAN,MXORAD,MXORD)
CALL TRFVAL(NTFA,TFCA,1.0,MXORAN,MXORAD)

NOW SAVE THE FOLLOWING DATA
VARIABLES MXORD,MXGRSN,MXORSO,MXCRAN,MXORAD,NTFS,NTFA
ARRAYS TFCS,TMSS,PHISS,TFCA,TMSA,PHISA

RETURN
END

```

```

C CCCCCCCCC

```

```

ACSMOD54
ACSMOD55
ACSMOD56
ACSMOD57
ACSMOD58
ACSMOD59
ACSMOD60
ACSMOD61
ACSMOD62
ACSMOD63
ACSMOD64
ACSMOD65
ACSMOD66
ACSMOD67

```

AD-A106 520

DOUGLAS AIRCRAFT CO. LONG BEACH CA

F/O 18/3

NUCLEAR BLAST RESPONSE COMPUTER PROGRAM. VOLUME III. PROGRAM LI-ETC(U)

AUG 61 J A MCOREY, H H CROXEN, T P KALHAN

DNA001-75-C-0216

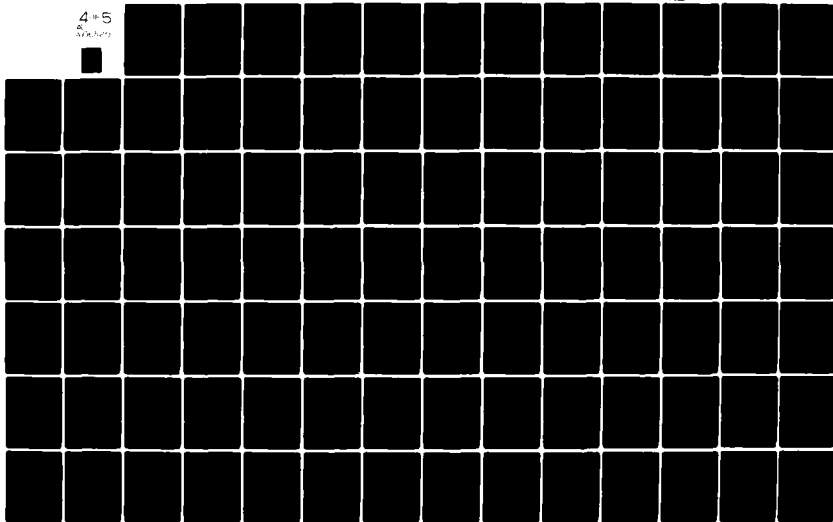
UNCLASSIFIED

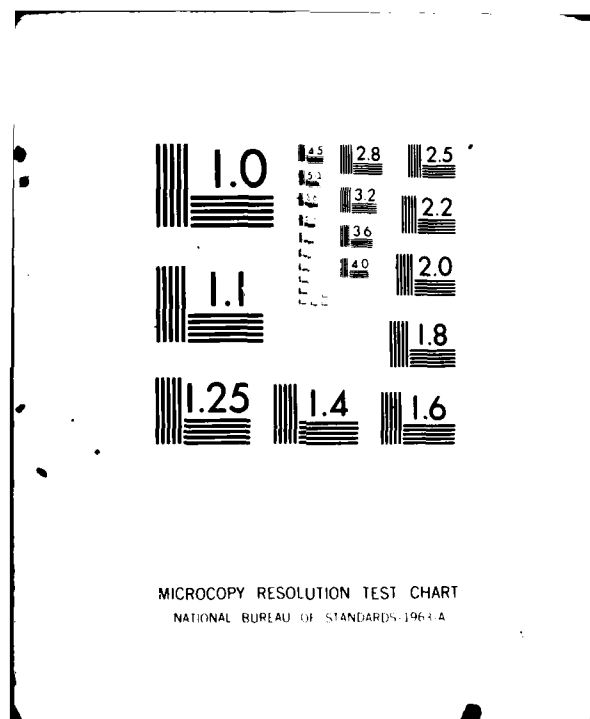
AFWL-TR-61-32-VOL-3

NL

4-5

4-5






```

SUBROUTINE TFLIN(NTI,
1 NTF, MXBLK, MXOBLK, ANUM, ADEN, NZER, FACT, NPRN)
C
C COMMON /ZZZ/CASE(34), TITLE(14), NIN, NOUT, <ROW, LINES, IPRNT, NER
C
C DIMENSION ANUM(NTF, MXBLK, 1)
C DIMENSION ADEN(NTF, MXBLK, 1)
C DIMENSION TN(3), TD(3), TL(3)
C DATA BLANK/
C
C 10 FORMAT(3F12.0, 3(3A4), I37, 3F12.0)
C 20 FORMAT(1H, 3(14, 2X), 3(4X, 3A4))
C 25 FORMAT(1H0, TRANSFER FUNCTION BLOCK/ELEMENT POLYNOMIAL INPUT /1H0,
1 IFNO BLKNO ORDER, 12X, NUM, 12X, DEN, 6X, NTFL)
C
C IF (FACT.EQ.0) FACT=1.0
C IF (NZER.NE.0) GO TO 90
C IF (NTF.NE.MXBLK) MXOBLK
C 90 I=1, IFIN
C ANUM(I, 1, 1)=0.
C ADEN(I, 1, 1)=0.
C
C CONTINUE
C
C NTFN = TF NO.
C NCRD = ORDER OF ELEMENT IN BLOCK
C NBLK = BLOCK NO.
C AN = NUM COEF
C AD = DEN COEF
C
C IF (NPRN.NE.0) WRITE(NOUT, 25)
C CONTINUE
C NTFL=0
C READ(NTI, 10) ATFN, ABLK, AORD, TN, TD, TL, AN, AD, TLAST
C NTFN=ATFN
C NBLK=ABLK
C NCRD=AORD
C IF (NTFN.LT.0) GO TO 144
C IF (NPRN.NE.0)
1 WRITE(NOUT, 20) NTFN, ABLK, NORD, TN, TD, TL
C NCRD=NORD+1
C IF (NTFN.GT. NTF) GO TO 730
C IF (NTFN.LT. 1) GO TO 730
C IF (NBLK.GT. MXOBLK) GO TO 725
C IF (NCRD.GT. MXOBLK) GO TO 735
C IF (TL(1).NE. BLANK.OR. TL(2).NE. BLANK.OR. TL(3).NE. BLANK) NTFL=TLAST
C IF (NTFL.NE.0) GO TO 120
C IF (TN(1).NE. BLANK.OR. TN(2).NE. BLANK.OR. TN(3).NE. BLANK)
1 ANUM(NTFN, NBLK, NCRD)=AN*FACT
C IF (TD(1).NE. BLANK.OR. TD(2).NE. BLANK.OR. TD(3).NE. BLANK)
1 ADEN(NTFN, NBLK, NCRD)=AD*FACT
C GO TO 100

```

TFBLIN 2
TFBLIN 3
TFBLIN 4
TFBLIN 5
TFBLIN 6
TFBLIN 7
TFBLIN 8
TFBLIN 9
TFBLIN 10
TFBLIN 11
TFBLIN 12
TFBLIN 13
TFBLIN 14
TFBLIN 15
TFBLIN 16
TFBLIN 17
TFBLIN 18
TFBLIN 19
TFBLIN 20
TFBLIN 21
TFBLIN 22
TFBLIN 23
TFBLIN 24
TFBLIN 25
TFBLIN 26
TFBLIN 27
TFBLIN 28
TFBLIN 29
TFBLIN 30
TFBLIN 31
TFBLIN 32
TFBLIN 33
TFBLIN 34
TFBLIN 35
TFBLIN 36
TFBLIN 37
TFBLIN 38
TFBLIN 39
TFBLIN 40
TFBLIN 41
TFBLIN 42
TFBLIN 43
TFBLIN 44
TFBLIN 45
TFBLIN 46
TFBLIN 47
TFBLIN 48
TFBLIN 49
TFBLIN 50
TFBLIN 51
TFBLIN 52
TFBLIN 53

TFBLIN54
TFBLIN55
TFBLIN56
TFBLIN57
TFBLIN58
TFBLIN59
TFBLIN60
TFBLIN61
TFBLIN62
TFBLIN63
TFBLIN64
TFBLIN65
TFBLIN66
TFBLIN67
TFBLIN68
TFBLIN69
TFBLIN70
TFBLIN71
TFBLIN72
TFBLIN73
TFBLIN74
TFBLIN75
TFBLIN76
TFBLIN77
TFBLIN78
TFBLIN79
TFBLIN80
TFBLIN81
TFBLIN82
TFBLIN83
TFBLIN84
TFBLIN85
TFBLIN86
TFBLIN87
TFBLIN88

```

C 120 CONTINUE
C PUT THIS TF BLOCK IN ALL DESIGNATED TF BLOCKS
C
  IF (NTFL.GT.NTF)NTFL=NTF
  DO 130 N=NTFN,NTFL
  IF (TN(1).NE.BLANK.OR.TN(2).NE.BLANK.OR.TN(3).NE.BLANK)
  1ANUM(N,NBLK,NORD)=AN*FACT
  IF (TD(1).NE.BLANK.OR.TD(2).NE.BLANK.OR.TD(3).NE.BLANK)
  1ADFN(N,NBLK,NORD)=AD*FACT
  CONTINUE
  GO TO 100
C 144 CONTINUE
C
  RETURN
C 735 CONTINUE
  WRITE (NGOUT,735)
  736 ECRMAT(1H,CPDER OF INPUT ELEMENT EXCEEDS MAX - SOLN HALTS )
  STOP
C 730 CONTINUE
  731 WRITE (NGOUT,731)
  ECRMAT(1H,CPDER OF INPUT TF NO EXCEEDS MAX OR IS 0 - SOLN PROCEEDS )
  GO TO 100
C 725 CONTINUE
  726 WRITE (NGOUT,726)
  ECRMAT(1H,CPDER OF INPUT BLOCK NO EXCEEDS MAX - SOLN PROCEEDS )
  GO TO 100
C
  END

```

```

C 10 SUBROUTINE TFBLOK(
21 1 NTF,MXBLK,MXOBLK,TFC,ANUM,ADEN,ATFN,ATFD, ANUMT,ADENT,
20 1 MXORDN,MXORDD,NORDM)
30 COMMON /ZZZ/CASE(34),TITLE(14),NIN,NOUT,<ROW,LINES,IPRNT,NER
C 21
20 DIMENSION TFC(2,NTF,1)
30 DIMENSION ANUM(NTF,MXBLK,1)
40 DIMENSION ADEN(NTF,MXBLK,1)
45 DIMENSION ATFN(2,1)
C 21 DIMENSION ATFD(2,1)
20 DIMENSION ANUMT(MXBLK,1)
30 DIMENSION ADENT(MXBLK,1)
C 21
20 FORMAT(1H, 3(14,2X),16X,E16.8)
30 FORMAT(1H, 3(14,2X),E16.8)
40 FORMAT(1H, TRANSFER FUNCTION POLYNOMIAL /1H0,
45 1 TFND ORDER CODE ,11X,REAL,12X,IMAG)
C 21 1 FORMAT(1H, SUMMARY ALL BLOCK INPUT TF DATA /1H0,
20 1 3( TFNBLKORD ,4X, NUM ,8X, DEN ,4X))
30 1 FORMAT(1H ,3(313,1X,2E11.3))
C 21
20 CODE= 1, NUM COEF , = 2, DEN COEF
30 IF (NZER.NE.0) GO TO 90
40 MXORDN=0
45 MXCRDD=0
C 21
20 CONTINUE
30 IF (IPRNT.EQ.0) GO TO 120
40 CALL HEADNG
45 WRTTF(NOUT,40)
C 21 ICL=5
20 NF=1
30 NL=3
40 CONTINUE
45 IF (NL.GT.NTF) NL=NTF
C 101 DO 110 J=1,MXBLK
101 DO 110 K=1,MXOBLK
105 KK=K-1
110 ICL=ICL+1
115 IF (ICL.LT.KROW) GO TO 105
120 CALL HEADNG
125 WRTTE(NOUT,40)
130 ICL=5
135 CONTINUE
140 WRTTE(NOUT,45) (N,J,KK,ANUM(N,J,K),ADEN(N,J,K),N=NF,NL)
145 WRTTF(NOUT,10)
150 ICL=ICL+1
155 IF (NL.GF.NTF) GO TO 120

```

TFBLOK 2
TFBLOK 3
TFBLOK 4
TFBLOK 5
TFBLOK 6
TFBLOK 7
TFBLOK 8
TFBLOK 9
TFBLOK 10
TFBLOK 11
TFBLOK 12
TFBLOK 13
TFBLOK 14
TFBLOK 15
TFBLOK 16
TFBLOK 17
TFBLOK 18
TFBLOK 19
TFBLOK 20
TFBLOK 21
TFBLOK 22
TFBLOK 23
TFBLOK 24
TFBLOK 25
TFBLOK 26
TFBLOK 27
TFBLOK 28
TFBLOK 29
TFBLOK 30
TFBLOK 31
TFBLOK 32
TFBLOK 33
TFBLOK 34
TFBLOK 35
TFBLOK 36
TFBLOK 37
TFBLOK 38
TFBLOK 39
TFBLOK 40
TFBLOK 41
TFBLOK 42
TFBLOK 43
TFBLOK 44
TFBLOK 45
TFBLOK 46
TFBLOK 47
TFBLOK 48
TFBLOK 49
TFBLOK 50
TFBLOK 51
TFBLOK 52
TFBLOK 53

TFBLOK54
TFBLOK55
TFBLOK56
TFBLOK57
TFBLOK58
TFBLOK59
TFBLOK60
TFBLOK61
TFBLOK62
TFBLOK63
TFBLOK64
TFBLOK65
TFBLOK66
TFBLOK67
TFBLOK68
TFBLOK69
TFBLOK70
TFBLOK71
TFBLOK72
TFBLOK73
TFBLOK74
TFBLOK75
TFBLOK76
TFBLOK77
TFBLOK78
TFBLOK79
TFBLOK80
TFBLOK81
TFBLOK82
TFBLOK83
TFBLOK84
TFBLOK85
TFBLOK86
TFBLOK87
TFBLOK88
TFBLOK89
TFBLOK90
TFBLOK91
TFBLOK92
TFBLOK93
TFBLOK94
TFBLOK95
TFBLOK96
TFBLOK97
TFBLOK98
TFBLOK99
TFBLO100
TFBLO101
TFBLO102
TFBLO103
TFBLO104
TFBLO105

```

NF=NL+1
NL=NL+3
GC TC 101
CONTINUE
120 C
CALL HEADNG
WRITE (NOUT,30)
ICL=5
C
NCRDMT=2*NORDM
DC 1000 N=1,NTF
C
TEST DEN FOR ZEROS
DC 600 J=1,MXRLK
ICHD=0
DC 500 K=1,MXCBLK
ANUMT(J,K)=ANUM(N,J,K)
ADENT(J,K)=ADEN(N,J,K)
IF (ADENT(J,K).NE.0) ICHKD=1
CONTINUE
500 IF (ICHD.EQ.0) ADENT(J,1)=1.0
600 CONTINUE
C
DC 145 J=1,NORDMT
ATFN(J,1)=0.
ATFD(J,1)=0.
ATFN(1,1)=1.0
ATFD(1,1)=1.0
C
DC 260 M=1,2
DC 260 K=1,NORDM
TEC(M,N,K)=0.
C
DC 160 J=1,MXBLK
DC 150 L=1,MXOBLK
KK=0
DC 150 K=L,NORDM
KK=KK+1
ATFN(2,K)=ATFN(2,K)+ANUMT(J,L)*ATFN(1,KK)
ATFD(2,K)=ATFD(2,K)+ADENT(J,L)*ATFD(1,KK)
CONTINUE
150 DC 155 K=1,NORDM
ATFN(1,K)=ATFN(2,K)
ATFD(1,K)=ATFD(2,K)
ATFN(2,K)=0.
ATFD(2,K)=0.
CONTINUE
155 C
160 C
C
DC 300 K=1,NORDM,4

```

```

IF (ATFN(1,K).NE.0.AND.K.GT.MXCRDN)MXORDN=K
IF (ATFD(1,K).NE.0.AND.K.GT.MXORDD)MXORDD=K
TFC(1,N,K)=ATFN(1,K)
TFC(2,N,K)=ATFD(1,K)
CONTINUE
C 300
DC 310 K=2,NORDM,4
IF (ATFN(1,K).NE.0.AND.K.GT.MXORDN)MXORDN=K
IF (ATFD(1,K).NE.0.AND.K.GT.MXORDD)MXORDD=K
TFC(1,N,K)=ATFN(1,K)
TFC(2,N,K)=ATFD(1,K)
CONTINUE
C 310
IF (NCRDM.LT.3)GO TO 350
DC 320 K=3,NORDM,4
IF (ATFN(1,K).NE.0.AND.K.GT.MXORDN)MXORDN=K
IF (ATFD(1,K).NE.0.AND.K.GT.MXORDD)MXORDD=K
TFC(1,N,K)=ATFN(1,K)
TFC(2,N,K)=ATFD(1,K)
CONTINUE
C 320
IF (NCRDM.LT.4)GO TO 350
DC 330 K=4,NORDM,4
IF (ATFN(1,K).NE.0.AND.K.GT.MXORDN)MXORDN=K
IF (ATFD(1,K).NE.0.AND.K.GT.MXORDD)MXORDD=K
TFC(1,N,K)=ATFN(1,K)
TFC(2,N,K)=ATFD(1,K)
CONTINUE
C 330
CONTINUE
C 350
IF (IPRNT.EQ.0)GO TO 1000
L=1
DC 400 K=1,MXORDN,2
KK=K-1
ICL=ICL+1
IF (ICL.LT.KROW)GO TO 395
CALL HEADNG
WRITE (NOUT,30)
ICL=5
CONTINUE
C 395
WRITE (NOUT,20)N,KK,L,TFC(L,N,K)
WRITE (NOUT,21)N,K,L,TFC(L,N,K+1)
CONTINUE
C 400
L=2
DC 410 K=1,MXORDD,2
KK=K-1
ICL=ICL+1
IF (ICL.LT.KROW)GO TO 405
CALL HEADNG
WRITE (NOUT,30)
ICL=5
CONTINUE
C 405

```

```

TF8L0106
TF8L0107
TF8L0108
TF8L0109
TF8L0110
TF8L0111
TF8L0112
TF8L0113
TF8L0114
TF8L0115
TF8L0116
TF8L0117
TF8L0118
TF8L0119
TF8L0120
TF8L0121
TF8L0122
TF8L0123
TF8L0124
TF8L0125
TF8L0126
TF8L0127
TF8L0128
TF8L0129
TF8L0130
TF8L0131
TF8L0132
TF8L0133
TF8L0134
TF8L0135
TF8L0136
TF8L0137
TF8L0138
TF8L0139
TF8L0140
TF8L0141
TF8L0142
TF8L0143
TF8L0144
TF8L0145
TF8L0146
TF8L0147
TF8L0148
TF8L0149
TF8L0150
TF8L0151
TF8L0152
TF8L0153
TF8L0154
TF8L0155
TF8L0156
TF8L0157

```

TF8L0158
 TF8L0159
 TF8L0160
 TF8L0161
 TF8L0162
 TF8L0163
 TF8L0164
 TF8L0165
 TF8L0166
 TF8L0167
 TF8L0168
 TF8L0169

```

      WRITE (NOUT,20) N,K,K,L,TFC(L,N,K)
      WRITE (NOUT,21) N,K,L,TFC(L,N,K+1)
      CONTINUE
C
      WRITE (NOUT,10)
      ICL=ICL+1
C
      CONTINUE
C
      CONTINUE
      RETURN
      END
  
```

```

C          SUBROUTINE TRFVAL(NTF,A ,OMEGA,MXORDN,MXORDD)
C          COMMON /ZZZ/CASE(34),TITLE(14),NIN,NOUT,KROW,LINES,IPRNT,NER
C          DIMENSION A(2,NTF,1)
C          50 FORMAT(1H0, TRANSFER FUNCTIONS FOR 1 RAD/SEC /1H0,
1          1 7X, RE(NTF), 6X, IM(NUM), 6X, RE(DEN), 6X, IM(DEN) ,
2          2 7X, RE(TF), 7X, IM(TF), 10X, MOD, 8X, PHASE )
C          60 FORMAT(1H ,14 ,2X,7E13.5,5X,F8.3)
C          WRITE(NOUT,50)
DO 220 N=1,NTF
AR=0.
AI=0.
BR=1.
BI=0.
IF (MXORDN.EQ.0)GO TO 600
DO 300 I=1,MXORDN,2
NORD=I-1
POWR=OMEGA**NORD
PCWI=OMEGA**I
AP=AR+A(1,N,I)*POWR
AI=AI+A(1,N,I+1)*POWI
300 CONTINUE
IF (MXORDD.EQ.0)GO TO 600
BR=0.
DO 500 I=1,MXORDD,2
NORD=I-1
POWR=OMEGA**NORD
PCWI=OMEGA**I
BR=BR+A(2,N,I)*POWR
BI=BI+A(2,N,I+1)*POWI
500 CONTINUE
C          CONTINUE
C          600
C          DEN=BR**2+BI**2
C          TFR=(AR*BR+AI*BI)/DEN
C          TFI=(AI*BR-AR*BI)/DEN
C          DEN=SQRT(TFR**2+TFI**2)
C          TAN=0.0
C          IF (TFR.NE.0)GO TO 200
C          IF (TFI.LT.0)TAN=-1.57080
C          IF (TFI.GT.0)TAN= 1.57080
C          GO TO 210
C          200 CONTINUE
C          TAN=ATAN2(TFI,TFR)
C          210 CONTINUE
C          TAN=TAN*57.29578
C          WRITE(NOUT,60)N,AR ,AI ,BR ,BI ,TFR,TFI,DEN,TAN

```

```

TRF VAL 2
TRF VAL 3
TRF VAL 4
TRF VAL 5
TRF VAL 6
TRF VAL 7
TRF VAL 8
TRF VAL 9
TRF VAL 10
TRF VAL 11
TRF VAL 12
TRF VAL 13
TRF VAL 14
TRF VAL 15
TRF VAL 16
TRF VAL 17
TRF VAL 18
TRF VAL 19
TRF VAL 20
TRF VAL 21
TRF VAL 22
TRF VAL 23
TRF VAL 24
TRF VAL 25
TRF VAL 26
TRF VAL 27
TRF VAL 28
TRF VAL 29
TRF VAL 30
TRF VAL 31
TRF VAL 32
TRF VAL 33
TRF VAL 34
TRF VAL 35
TRF VAL 36
TRF VAL 37
TRF VAL 38
TRF VAL 39
TRF VAL 40
TRF VAL 41
TRF VAL 42
TRF VAL 43
TRF VAL 44
TRF VAL 45
TRF VAL 46
TRF VAL 47
TRF VAL 48
TRF VAL 49
TRF VAL 50
TRF VAL 51
TRF VAL 52
TRF VAL 53

```

TRF VAL54
TRF VAL55
TRF VAL56
TRF VAL57

C 220 CONTI NUE
RE TURN
END

CFRQ R 2
CFRQ R 3
CFRQ R 4
CFRQ R 5
CFRQ R 6
CFRQ R 7
CFRQ R 8
CFRQ R 9
CFRQ R 10
CFRQ R 11
CFRQ R 12
CFRQ R 13
CFRQ R 14
CFRQ R 15
CFRQ R 16
CFRQ R 17
CFRQ R 18
CFRQ R 19
CFRQ R 20
CFRQ R 21
CFRQ R 22
CFRQ R 23
CFRQ R 24
CFRQ R 25
CFRQ R 26
CFRQ R 27
CFRQ R 28
CFRQ R 29
CFRQ R 30
CFRQ R 31
CFRQ R 32
CFRQ R 33
CFRQ R 34
CFRQ R 35
CFRQ R 36
CFRQ R 37
CFRQ R 38

```

SUBROUTINE COEFF(RKIN,RK,NK,C,COEFF)
DIMENSION RK(1),C(13,10),COEF(1)
REAL KF(13)

C *** FORM KF
      NK3 = NK+3
      KF(1) = 1.0
      KF(2) = RKIN
      KF(3) = 0.0
      DO 10 J=4,NK3
        A = ( RKIN-RK(J-3) ) **2
        10 KF(J) = A*ALOG(A)

C
      DO 20 J=1,NK
        COEF(J) = 0.0
      DO 20 I=1,NK3
        20 COEF(J) = COEF(J) + KF(I) * C(I,J)

C
      RETURN
      END

```

```

SUBROUTINE FORMC (RK,NK,W,IPRN)
DIMENSION RK(1),W(13,10)
REAL KD(13,13)
NK3 = NK+3
DO 30 I=1,NK3
DO 10 J=1,NK3
10 KD(I,J) = 0.0
20 DO 20 J=1,NK
20 W(I,J) = 0.0
30 CONTINUE
DO 40 J=1,NK
I = J+3
40 W(I,J) = 1.0
KC(3,3) = 1.0
DO 60 J=4,NK3
KC(1,J) = 1.0
KC(J,1) = 1.0
KC(J,2) = RK(J-3)
KC(J,2) = RK(J-3)
DO 50 I=4,NK3
IF (I.EQ.J) GO TO 50
A = ( RK(I-3)-RK(J-3) ) **2
KD(I,J) = A*ALOG(A)
KD(J,I) = KD(I,J)
50 CONTINUE
60 KD(J,J) = 0.0
D = 0.0
CALL MISC(KD,NK3,13,W,NK,NERR,D)
IF (IPRN.GT.1)
1 CALL PRINT(NK3,NK,13,1,8HW MATRIX,2)
RETURN
END

```

```

FORMC 2
FORMC 3
FORMC 4
FORMC 5
FORMC 6
FORMC 7
FORMC 8
FORMC 9
FORMC 10
FORMC 11
FORMC 12
FORMC 13
FORMC 14
FORMC 15
FORMC 16
FORMC 17
FORMC 18
FORMC 19
FORMC 20
FORMC 21
FORMC 22
FORMC 23
FORMC 24
FORMC 25
FORMC 26
FORMC 27
FORMC 28
FORMC 29
FORMC 30
FORMC 31
FORMC 32
FORMC 33

```

```

SUBROUTINE FREQRS(
1  CP,AMCNDNO,FREQIN,EMBAR,CAY,CXPDPG,NMD
2  D,F,OMEGA,Q,
3  NVBW,TFM,TFC,TMSS,TMSSA,PHISS,PHISA,NTFS,NTFA,
6  MXORSN,MXORAN,MXORSO,MXORAD,
7  SPLS,SPLA,NBOX,NSBETO)
C
COMMON NAA,A(1)
C
COMMON/ZZZ/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
COMMON/XTF/NFLAGS(100)
C
EQUIVALENCE (NFLAGS(6),NSYM),(NFLAGS(7),NASYM)
EQUIVALENCE (NFLAGS(11),IDENT)
EQUIVALENCE (NFLAGS(12),BZERO), (NFLAGS(13),RHOO)
1 , (NFLAGS(14),STGMA ), (NFLAGS(15),EQUVAS), (NFLAGS(16),IUNITS)
2 , (NFLAGS(17),VKEAS), (NFLAGS(18),VTFPS)
3 , (NFLAGS(20),KPRCHK), (NFLAGS(21),IPREQ )
4 , (NFLAGS(19),ALT)
5 , (NFLAGS(100),SIZECT)
6 , (NFLAGS(42),NG)
7 , (NFLAGS(1),AMACH)
C
DIMENSION IHD(50), RHD(50)
EQUIVALENCE (IHD(1),RHD(1))
DIMENSION UNITS(3)
DIMENSION VELUN(3)
DIMENSION CR(3,8)
DIMENSION AMCNDNO(20)
DIMENSION FREQIN(1)
DIMENSION EMBAR(NMD,NMD)
DIMENSION CAY(1)
DIMENSION CXPDPG(1)
DIMENSION D(2,1)
DIMENSION F(2,1)
DIMENSION SPLS(1),SPLA(1)
DIMENSION Q(2,NMD)
DIMENSION OMEGA(1)
DIMENSION TFM(1)
DIMENSION TFC(1),TFCAL(1),PHISS(1),PHISA(1)
DIMENSION TMSS(NTFS,1),TMSSA(NTFA,1)
DIMENSION RK(10),C(13,10)
DIMENSION COEF(10),VORW(10),IVRW(10)
C
COMMON/DISK2/ND2,ITBL2(843),NRECSA,IBUMP,VKD,VORWS(20)
COMMON/AFROMX/INTARC,NVBWMX,VORWIN(400),RINTP(50,3)
C
20 FORMAT(1H0,20X,GUST ORIENTATION NO. ,I4/1H ,
1 20X,GAMX = ,F8.5/1H ,
1 20X,GAM7 = ,F8.5/1H ,
30 FORMAT(1H ,I4,2E16.8)

```

```

FREQRS 2
FREQRS 3
FREQRS 4
FREQRS 5
FREQRS 6
FREQRS 7
FREQRS 8
FREQRS 9
FREQRS 10
FREQRS 11
FREQRS 12
FREQRS 13
FREQRS 14
FREQRS 15
FREQRS 16
FREQRS 17
FREQRS 18
FREQRS 19
FREQRS 20
FREQRS 21
FREQRS 22
FREQRS 23
FREQRS 24
FREQRS 25
FREQRS 26
FREQRS 27
FREQRS 28
FREQRS 29
FREQRS 30
FREQRS 31
FREQRS 32
FREQRS 33
FREQRS 34
FREQRS 35
FREQRS 36
FREQRS 37
FREQRS 38
FREQRS 39
FREQRS 40
FREQRS 41
FREQRS 42
FREQRS 43
FREQRS 44
FREQRS 45
DISK2 2
FREQRS 47
FREQRS 48
FREQRS 49
FREQRS 50
FREQRS 51
FREQRS 52
FREQRS 53

```


FREORI106
 FREORI107
 FREORI108
 FREORI109
 FREORI110
 FREORI111
 FREORI112
 FREORI113
 FREORI114
 FREORI115
 FREORI116
 FREORI117
 FREORI118
 FREORI119
 FREORI120
 FREORI121
 FREORI122
 FREORI123
 FREORI124
 FREORI125
 FREORI126
 FREORI127
 FREORI128
 FREORI129
 FREORI130
 FREORI131
 FREORI132
 FREORI133
 FREORI134
 FREORI135
 FREORI136
 FREORI137
 FREORI138
 FREORI139
 FREORI140
 FREORI141
 FREORI142
 FREORI143
 FREORI144
 FREORI145
 FREORI146
 FREORI147
 FREORI148
 FREORI149
 FREORI150
 FREORI151
 FREORI152
 FREORI153
 FREORI154
 FREORI155
 FREORI156
 FREORI157

C NTOTAP = NBOX + 2*NSRETO
 C READ BACK GEN GUST INTEGRATION MATS
 C
 C NK=NTOTAP*NSYM
 C LCC=NKD*IRUMP+1
 C CALL READMS (ND2,SPLS,NW,LOC)
 C NW=NTOTAP*NASYM
 C LCC=LCC+1
 C CALL READMS (ND2,SPLA,NW,LOC)
 C
 C DO 240 IVRW=1,NVBW
 C FREQ=VOBWIN (IVRW)
 C OMEGA (IVRW)=TWOPI*FREQ
 C IF (OMEGA (IVRW).LE.O) OMEGA (IVRW)=1.E-06
 C 240 CONTINUE
 C
 C DO 190 I=1,50
 C IHD (1) = 0
 C IHD (2) = IDENT
 C IHD (3) = NG
 C IHD (4) = NVBW
 C IHD (5) = NSYM
 C IHD (6) = NASYM
 C IHD (7) = NK
 C IHD (8) = NTOTAP
 C RHD (11) = VTFPS
 C RHD (12) = SIGMA
 C CALL WFRSP (1,0,0,IHD,NOUT,NER)
 C CALL WFRSP (2,0,0,CR,NOUT,NER)
 C CALL WFRSP (3,0,0,OMEGA,NOUT,NER)
 C
 C L1S = 2*NSYM*NSYM*NK
 C L1A = 2*NASYM*NASYM*NK
 C L2S = 2*NTOTAP*NK
 C L2A = 2*NTOTAP*NK
 C L3S = NTOTAP*NK
 C L3A = NTOTAP*NK
 C L4S = 2*NTOTAP
 C L4A = 2*NTOTAP
 C N1S = NAA
 C N2S = N1S + L1S
 C N3S = N2S + L2S
 C N4S = N3S + L3S
 C N5S = N4S + L4S
 C N1A = N1S + L1A
 C N2A = N2S + L2A
 C N3A = N3S + L3A
 C N4A = N4S + L4A

FREQR210
 FREQR211
 FREQR212
 FREQR213
 FREQR214
 FREQR215
 FREQR216
 FREQR217
 FREQR218
 FREQR219
 FREQR220
 FREQR221
 FREQR222
 FREQR223
 FREQR224
 FREQR225
 FREQR226
 FREQR227
 FREQR228
 FREQR229
 FREQR230
 FREQR231
 FREQR232
 FREQR233
 FREQR234
 FREQR235
 FREQR236
 FREQR237
 FREQR238
 FREQR239
 FREQR240
 FREQR241
 FREQR242
 FREQR243
 FREQR244
 FREQR245
 FREQR246
 FREQR247
 FREQR248
 FREQR249
 FREQR250
 FREQR251
 FREQR252
 FREQR253
 FREQR254
 FREQR255
 FREQR256
 FREQR257
 FREQR258
 FREQR259
 FREQR260
 FREQR261

```

WRITE (NOUT,175)
LINES=LINES+2
DO 320 I=1,NM
  F(1,I)=0.
  F(2,I)=0.
  I1=N5
  I2=N5+2*NTOTAP-1
  DO 321 I=1,12
    A(I)=0.0
  GO TO 590
320 CONTINUE
C
C
  IPREQ=0
  KPRCHK=0
  IF (ITEMPK.NE.0.AND.IG.EQ.1.AND.IVBW.EQ.1) KPRCHK=1
  IF (IG.EQ.1.PRK1.OR.IG.EQ.1.PRK2) KPRCHK=1
  IF (IG.EQ.1.PRQ1.OR.IG.EQ.1.PRQ2) IPREQ=1
C
  IF (IND.EQ.3.AND.IZROA.EQ.0) GO TO 290
  IF (IND.EQ.1.AND.IZROS.EQ.0) GO TO 290
C
DO 210 I=1,NM
  F(1,I)=0.
  F(2,I)=0.
  I1=N5
  I2=N5+2*NTOTAP-1
  DO 211 I=1,12
    A(I)=0.0
  GO TO 590
210 CONTINUE
C
C
  CALL TOTINT(
    1 VOBW,ISETA,IMC,D,A(N1),A(N2),NTOTAP,IG,COEF,VM,NK,KPRCHK,IND,
    3 VOBWI,IVRWI,IPEG,RK,C)
C
  IPREQ=0
  KPRCHK=0
  IF (IG.EQ.1.PRQ1.OR.IG.EQ.1.PRQ2) IPREQ=1
  IF (IG.EQ.1.PRK1.OR.IG.EQ.1.PRK2) KPRCHK=1
C
C
  ICACS=0
  IF (IND.EQ.3) GO TO 331
  IF (INTFS.EQ.0) GO TO 332
  ICACS=1
  CALL TFMATF (INTFS,NM,TFM,D,OMEGA(IVBW),TFCS,IMSS,0,
    1 MXORSN,MXORSN,PHISS,KPRCHK)
  GO TO 332
331 IF (INTFA.EQ.0) GO TO 332
  
```

```

C 332 CONTINUE
C
C 333 CONTINUE
C
C 340 CONTINUE
C
C 345
C 346
C 350 CONTINUE
C
C 400
C
C 405
C
C 410
C
C 415
C
C 420
C
C 425
C
C 430
C
C 435
C
C 440
C
C 445
C
C 450
C
C 455
C
C 460
C
C 465
C
C 470
C
C 475
C
C 480
C
C 485
C
C 490
C
C 495
C
C 500
C
C 505
C
C 510
C
C 515
C
C 520
C
C 525
C
C 530
C
C 535
C
C 540
C
C 545
C
C 550
C
C 555
C
C 560
C
C 565
C
C 570
C
C 575
C
C 580
C
C 585
C
C 590
C
C 595
C
C 600
C
C 605
C
C 610
C
C 615
C
C 620
C
C 625
C
C 630
C
C 635
C
C 640
C
C 645
C
C 650
C
C 655
C
C 660
C
C 665
C
C 670
C
C 675
C
C 680
C
C 685
C
C 690
C
C 695
C
C 700
C
C 705
C
C 710
C
C 715
C
C 720
C
C 725
C
C 730
C
C 735
C
C 740
C
C 745
C
C 750
C
C 755
C
C 760
C
C 765
C
C 770
C
C 775
C
C 780
C
C 785
C
C 790
C
C 795
C
C 800
C
C 805
C
C 810
C
C 815
C
C 820
C
C 825
C
C 830
C
C 835
C
C 840
C
C 845
C
C 850
C
C 855
C
C 860
C
C 865
C
C 870
C
C 875
C
C 880
C
C 885
C
C 890
C
C 895
C
C 900
C
C 905
C
C 910
C
C 915
C
C 920
C
C 925
C
C 930
C
C 935
C
C 940
C
C 945
C
C 950
C
C 955
C
C 960
C
C 965
C
C 970
C
C 975
C
C 980
C
C 985
C
C 990
C
C 995
C

```

```

FREQR262
FREQR263
FREQR264
FREQR265
FREQR266
FREQR267
FREQR268
FREQR269
FREQR270
FREQR271
FREQR272
FREQR273
FREQR274
FREQR275
FREQR276
FREQR277
FREQR278
FREQR279
FREQR280
FREQR281
FREQR282
FREQR283
FREQR284
FREQR285
FREQR286
FREQR287
FREQR288
FREQR289
FREQR290
FREQR291
FREQR292
FREQR293
FREQR294
FREQR295
FREQR296
FREQR297
FREQR298
FREQR299
FREQR300
FREQR301
FREQR302
FREQR303
FREQR304
FREQR305
FREQR306
FREQR307
FREQR308
FREQR309
FREQR310
FREQR311
FREQR312
FREQR313

```

295

FREQR366
 FREQR367
 FREQR368
 FREQR369
 FREQR370
 FREQR371
 FREQR372
 FREQR373
 FREQR374
 FREQR375
 FREQR376
 FREQR377
 FREQR378
 FREQR379
 FREQR380
 FREQR381
 FREQR382
 FREQR383
 FREQR384
 FREQR385
 FREQR386
 FREQR387
 FREQR388
 FREQR389
 FREQR390
 FREQR391
 FREQR392
 FREQR393
 FREQR394
 FREQR395
 FREQR396
 FREQR397
 FREQR398
 FREQR399
 FREQR400
 FREQR401
 FREQR402
 FREQR403
 FREQR404
 FREQR405
 FREQR406
 FREQR407
 FREQR408
 FREQR409
 FREQR410
 FREQR411
 FREQR412
 FREQR413
 FREQR414
 FREQR415
 FREQR416
 FREQR417

```

C
M1=8
M2=9
IF(IIND.EQ.1)GO TO 415
M1=19
M2=20
415 DO 425 K=M1,M2
      I=AMODND(K)-M
      IF(I.LE.0)GO TO 425
      F(1,I)=0.
      F(2,I)=0.
      DO 420 J=1,NM
        LCC=I+(J-1)*NM
        D(1,LCC)=0.
        D(2,LCC)=0.
        LCC=J+(I-1)*NM
        D(1,LCC)=0.
        D(2,LCC)=0.
      CONTINUE
      LCC=I+(I-1)*NM
      D(1,LCC)=1.0
      CONTINUE
      C
      C
      IF(FREQ.GT.0.3)GO TO 588
      REDUCE OUT SING SOLN ELEM FOR LOW FREQ MATS
      C
      C
      M1=1
      M2=0
      IF(IIND.EQ.1)GO TO 575
      M1=11
      M2=0
      575 I=AMODND(M1)-M
      IF(I.LE.0)GO TO 580
      F(1,I)=0.
      F(2,I)=0.
      DO 576 J=1,NM
        LCC=I+(J-1)*NM
        D(1,LCC)=0.
        D(2,LCC)=0.
        LCC=J+(I-1)*NM
        D(1,LCC)=0.
        D(2,LCC)=0.
      CONTINUE
      LCC=I+(I-1)*NM
      D(1,LCC)=1.0
      CONTINUE
      576
      580 IF(IIND.EQ.3)GO TO 582
      IF(IIND.EQ.1.AND.M2.GT.3)GO TO 585
      M1=3
      M2=1
      GO TO 575
  
```

```

FREOR418
FREOR419
FREOR420
FREOR421
FREOR422
FREOR423
FREOR424
FREOR425
FREOR426
FREOR427
FREOR428
FREOR429
FREOR430
FREOR431
FREOR432
FREOR433
FREOR434
FREOR435
FREOR436
FREOR437
FREOR438
FREOR439
FREOR440
FREOR441
FREOR442
FREOR443
FREOR444
FREOR445
FREOR446
FREOR447
FREOR448
FREOR449
FREOR450
FREOR451
FREOR452
FREOR453
FREOR454
FREOR455
FREOR456
FREOR457
FREOR458
FREOR459
FREOR460
FREOR461
FREOR462
FREOR463
FREOR464
FREOR465
FREOR466
FREOR467
FREOR468
FREOR469

```

```

582 IF (M2.GT.0)GO TO 585
    M1=13
    M2=1
    GO TO 575
C
585 CONTINUE
C
588 CONTINUE
C
    IF (KPRCHK.EQ.0)GO TO 570
    WRITE (NOUT,60)IVBW,FREQ
    CALL PRNT(D,NM,NM,2,4H Z ,1)
    CALL PRNT(F,NM,1,NM,2,4H F ,1)
    LINES = LINES+1
570 CONTINUE
C
    DETDP = 0
C
    CALC Z-1 FOR GUST OR FORCE DRIVEN SYSTEM
C
    CALL MIS2(D,NM,NM,F,1,NER,DETD)
590 CONTINUE
C
    M1 = M+1
    M2 = M+NM
    N = 0
    DO 600 I=M1,M2
        N = N+1
        Q(1,I)=F(1,N)
        Q(2,I)=F(2,N)
        IF (IND.EQ.3)GO TO 601
        IMOS=IM0
        IREGS=IREG
        GO TO 602
    601 IMOA=IM0
        IREGA=IREG
        CONTINUE
C
    IF (NASYM.EQ.0) GO TO 700
    IF (IND.EQ.3) GO TO 700
    NM = NASYM
    M = NSYM
    IND = 3
    N1 = N1A
    N2 = N2A

```

FREQR470
FREQR471
FREQR472
FREQR473
FREQR474
FREQR475
FREQR476
FREQR477
FREQR478
FREQR479
FREQR480
FREQR481
FREQR482
FREQR483
FREQR484
FREQR485
FREQR486
FREQR487
FREQR488
FREQR489
FREQR490
FREQR491
FREQR492
FREQR493
FREQR494
FREQR495
FREQR496
FREQR497
FREQR498
FREQR499
FREQR500
FREQR501
FREQR502
FREQR503
FREQR504
FREQR505
FREQR506
FREQR507
FREQR508
FREQR509
FREQR510
FREQR511
FREQR512
FREQR513
FREQR514
FREQR515
FREQR516
FREQR517
FREQR518
FREQR519
FREQR520
FREQR521

```

N3 = N3A
N4 = N4A
N5 = N5A
IMO = IMOA
IFCG=IRECGA
GO TO 300

C 700 CONTINUE
C
C      SYMMETRIC AND ANTI-SYMMETRIC ANALYSES DONE
C
C      LINES=LINES+1
C      IF (IPREC.FQ.0) GO TO 820
C      IF (LINES.LE.KROW) GO TO 880
C      CALL HEADNG
C      WRITE(NOUT,20) IG, (CR(L,IG),L=1,3)
C      WRITE(NOUT,50) VKEAS,VELUN(IUNIT),DYNP,VWG,ALT
C      LINES = LINES+7
C      CONTINUE
C      880 WRITE(NOUT,40) IVBW,VORW,FREQ
C      DC 890 JJ=1,NMD
C      LINES=LINES+1
C      IF (LINES.LE.KROW) GO TO 895
C      CALL HEADNG
C      WRITE(NOUT,20) IG, (CR(L,IG),L=1,3)
C      WRITE(NOUT,50) VKEAS,VELUN(IUNIT),DYNP,VWG,ALT
C      LINES = LINES+7
C      CONTINUE
C      895 WRITE(NOUT,30) JJ,Q(1,JJ),Q(2,JJ)
C      CONTINUE
C      GO TO 850
C
C      820 IF (LINES.LE.KROW) GO TO 840
C      CALL HEADNG
C      WRITE(NOUT,20) IG, (CR(L,IG),L=1,3)
C      WRITE(NOUT,50) VKEAS,VELUN(IUNIT),DYNP,VWG,ALT
C      LINES=LINES+7
C      CONTINUE
C      840 WRITE(NOUT,45) IVBW,VORW,FREQ
C      CONTINUE
C
C      850 CONTINUE
C
C      IHD(1)=IG
C      IHD(2)=IVBW
C      IHD(4)=I ZROS
C      IHD(5)=I ZROA
C      CALL WRFRSP(4,IG,IVBW,IHD,NOUT,NER)
C      CALL WRFRSP(5,IG,IVBW,COEF,NOUT,NER)
C      CALL WRFRSP(5,IG,IVBW,IVBW,I,NOUT,NER)
C      CALL WRFRSP(7,IG,IVBW,Q,NOUT,NER)

```

```

C      CALL WRFERSP(8,IG,IVBW,A(N5S),NOUT,NER)
C      CALL WRFERSP(9,IG,IVBW,A(N5A),NOUT,NER)
C      800 CONTINUE
C      THIS ENDS FRFQ LOOP
C      IF (ICACS.NE.0) WRITE(NOUT,10)
C      900 CONTINUE
C      THIS ENDS GUST LOOP
C      IPRFQ=ITFMPQ
C      KPRCHK=ITE MPK
C      RETURN
C      1090 CONTINUE
C      WRITE(NOUT,100)
C      STOP
C      END

```

```

FREQR522
FREQR523
FREQR524
FREQR525
FREQR526
FREQR527
FREQR528
FREQR529
FREQR530
FREQR531
FREQR532
FREQR533
FREQR534
FREQR535
FREQR536
FREQR537
FREQR538
FREQR539
FREQR540
FREQR541
FREQR542
FREQR543

```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

SUBROUTINE GSTINI(IIZRO,FGUST,
1 SPLHP,NMODE,COEF,
2 F,FI,NM,NK,RS,TANS,IIECG,IPREQ,J)
COMMON/ZZZ/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
DIMENSION F(2,1)
DIMENSION FI(2,NM,1)
DIMENSION COEF(1)
DIMENSION RS(NM,1),TANS(NM,1)
DIMENSION FGUST(2,1)
DIMENSION SPLHP(NM,1)
THIS SUBROUTINE USES THE TRANSPOSE OF THE ABOVE MATRICES
DATA EP/1.0E-02/
NM = TOTAL AERO ELEMENTS = NBOX+2*NSBETO
10 FORMAT(IH0,20X,GUST INTERPOLATION ELEMENTS /IH0,
1 ELEM NG NK,12X,RR,12X,RI,11X,MOD,11X,TAN )
9 FORMAT(IH0,20X,SPLH MATRICES )
20 FORMAT(IH,3I4,4E14.6)
DO 45 I=1,NMODE
F(1,I)=0.
F(2,I)=0.
45 K=0
DO 40 I=1,NK
IF(COEF(I).EQ.1.0)K=I
40 CONTINUE
IF(K.NE.0)GO TO 100
IF(IIECG.NE.0)GO TO 120
ICNT=0
SET UP COUNT FOR ZERO GUST LOADS
IF(IIPREQ.EQ.0)GO TO 47
CALL HEADNG
WRITE (NOUT,9)
CALL PRNT(SPLHP,NM,NMODE,NM,1,4HSPLH,1)
CALL HEADNG
WRITE (NOUT,10)
C
  
```

GSTI NI 54
 GSTI NI 55
 GSTI NI 56
 GSTI NI 57
 GSTI NI 58
 GSTI NI 59
 GSTI NI 60
 GSTI NI 61
 GSTI NI 62
 GSTI NI 63
 GSTI NI 64
 GSTI NI 65
 GSTI NI 66
 GSTI NI 67
 GSTI NI 68
 GSTI NI 69
 GSTI NI 70
 GSTI NI 71
 GSTI NI 72
 GSTI NI 73
 GSTI NI 74
 GSTI NI 75
 GSTI NI 76
 GSTI NI 77
 GSTI NI 78
 GSTI NI 79
 GSTI NI 80
 GSTI NI 81
 GSTI NI 82
 GSTI NI 83
 GSTI NI 84
 GSTI NI 85
 GSTI NI 86
 GSTI NI 87
 GSTI NI 88
 GSTI NI 89
 GSTI NI 90
 GSTI NI 91
 GSTI NI 92
 GSTI NI 93
 GSTI NI 94
 GSTI NI 95
 GSTI NI 96
 GSTI NI 97
 GSTI NI 98
 GSTI NI 99
 GSTI NI 100
 GSTI NI 101
 GSTI NI 102
 GSTI NI 103
 GSTI NI 104
 GSTI NI 105

```

47  CONTINUE
    DC 90 I=1,NM
    DO 80 K=1,NK
      RD=FI(1,I,K)
      RI=FI(2,I,K)
      IF (ABS(RR).LT.EP) RR=0.
      IF (ABS(PI).LT.EP) RI=0.
      RT=SQRT(RR**2+RI**2)
      RS(I,K)=RR
      TANS(I,K)=RI
      IF (IPREQ.NE.0) WRITE(NOUT,20) I,J,K,RR,RI,RT
      IF (RR.NE.0) GO TO 80
      IF (RI.NE.0) GO TO 80
      IF (K.GE.2) GO TO 85
      GO TO 80
    C 80 CONTINUE
    GC IC 90
    85 CONTINUE
    DO 86 K=1,NK
      TANS(I,K)=0.
      RS(I,K)=0.
      ICNT=ICNT+NK
    C 86
    C 90 CONTINUE
    IFCG=1
    C
    IF (ICNT.GE.NK*NM) IZRO=1
    C
    IF (IZPO.EQ.1) GO TO 150
    GO TO 120
    C 100 CONTINUE
    DC 110 I=1,NM
    C
    FREAL=FI(1,I,K)
    FIMAG=FI(2,I,K)
    FGUST(1,I)=FREAL
    FGUST(2,I)=FIMAG
    C
    DO 182 L=1,NMODE
      F(1,L)=F(1,L)+SPLHP(I,L)*FREAL
      F(2,L)=F(2,L)+SPLHP(I,L)*FIMAG
    182 CONTINUE
    C 110 CONTINUE
    C
    GC TO 150
    C
  
```

GSTI N106
 GSTI N107
 GSTI N108
 GSTI N109
 GSTI N110
 GSTI N111
 GSTI N112
 GSTI N113
 GSTI N114
 GSTI N115
 GSTI N116
 GSTI N117
 GSTI N118
 GSTI N119
 GSTI N120
 GSTI N121
 GSTI N122
 GSTI N123
 GSTI N124
 GSTI N125
 GSTI N126
 GSTI N127
 GSTI N128
 GSTI N129
 GSTI N130
 GSTI N131
 GSTI N132
 GSTI N133
 GSTI N134
 GSTI N135

```

120 CONTINUE
DC 140 I=1,NM
R=0.
THE TA=0.
DC 130 K=1,NK
R=R+COEF(K)*RS(I,K)
130 THE TA=THE TA+COEF(K)*TANS(I,K)
C
FREAL=R
FIMAG=THE TA
FGUST(1,I)=FREAL
FGUST(2,I)=FIMAG
C
DC 282 L=1,NMCOE
F(1,L)=F(1,L)+SPLHP(I,L)*FREAL
F(2,L)=F(2,L)+SPLHP(I,L)*FIMAG
282 CONTINUE
C 140 CONTINUE
C 150 CONTINUE
C
IF(I.PREQ.EQ.0)GO TO 160
CALL PRNT(F,NMCOE,1,NMCOE,2,24HINTERP. GEN. GUST FORCE ,6)
C 160 CONTINUE
C
RETURN
C
END
  
```

```

SUBROUTINE INTERP(
1  F,VORW,DALL,VCBWT,NM,NK,COFF,IPRN,IPECC,INDI,RK,C)
2  COMPLEX D(NM,NM)
3  COMPLEX DALL(NM,NM,1)
4  DIMENSION VCBWT(1)
5  DIMENSION COFF(1)
6  DIMENSION RK(10),C(13,10)
7
8  DATA FPRV/0.010/
9  DATA EPC/0.10E-05/
10
11 IF(INDI IS SET 0, USE AVAILABLE COEFS
12 IF(INDI.EQ.0)GO TO 15
13
14 IF(INDI.EQ.0)GO TO 15
15
16 IF(ABS(1.-VORW(1))/VCBWT(1).LE.FPRV)II=1
17 RK(1)=1.0/VORW(1)
18 IF(IPECC.EQ.0)CALL FORMC(RK,NK,C,IPRN)
19 PKN=1.0/VCBWT
20 IF(II.NE.0)GO TO 30
21 CALL COEFF(RKIN,RK,NK,C,COFF)
22 DO 14 L=1,NK
23 IF(ABS(COFF(L)).LT.EPC)COFF(L)=0.0
24 CONTINUE
25 DO 20 I=1,NM
26 DO 20 J=1,NM
27 D(I,J)=0.
28 DO 20 L=1,NK
29 D(I,J)=COFF(L)*DALL(I,J,L)+D(I,J)
30 CONTINUE
31 DO 40 I=1,NM
32 DO 40 J=1,NM
33 D(I,J)=DALL(I,J,1)
34 COFF(1)=1.0
35 CONTINUE
36 IF(IPRN.EQ.0)GO TO 100
37 CALL PRNT(COFF,1,NK,1,1,4HCOEF,1)
38 CALL PRNT(D,NM,NM,2,20HDBAR(INTERPOLATED),5)
39 RETURN
40 END

```

```

INTERP 2
INTERP 3
INTERP 4
INTERP 5
INTERP 6
INTERP 7
INTERP 8
INTERP 9
INTERP10
INTERP11
INTERP12
INTERP13
INTERP14
INTERP15
INTERP16
INTERP17
INTERP18
INTERP19
INTERP20
INTERP21
INTERP22
INTERP23
INTERP24
INTERP25
INTERP26
INTERP27
INTERP28
INTERP29
INTERP30
INTERP31
INTERP32
INTERP33
INTERP34
INTERP35
INTERP36
INTERP37
INTERP38
INTERP39
INTERP40
INTERP41
INTERP42
INTERP43
INTERP44
INTERP45
INTERP46

```

```

SUBROUTINE TOTINT (VOBW, ISETA, IMO, D, DALL
1  , FALL, NTOTAP, IG, COEF, NM, NK, KPRCHK, IND,
3  VOBWI, IVRWI, IRECG, RK, C)
C
COMMON /ZZZ/CASE(49), NOUT, KROW, LINES
COMMON/DISK2/ND2, ITBL2(843), NRECSA, IBUMP, NKD, VOBWS(20)
C
DIMENSION D(1)
COMPLEX DALL(NM, NM, 1)
COMPLEX FALL(NTOTAP, NK)
DIMENSION COEF(1), VOBWI(1), IVRWI(1)
DIMENSION RK(10), C(13, 10)
C
M=1
IF (IND.EQ.3) M=2
NWD = 2*NM*NM
C
FIND OUT IF WE NEED NEW AERO
IRECC=0
TESTV=1.E06
DO 10 I=1, NKD
TEST=ABS(VORWS(I)-VOBWI)
IF (TEST.GT.TESTV) GO TO 10
TESTV=TEST
IM=I
10 CONTINUE
C
IL=IM-ISETA
IF (NK.GT.4) IL=IL-ISETA
IH=IL+(NK-1)*ISETA
C
20 IF (IL.GE.1.AND.ISETA.GT.0) GO TO 30
IF (IL.LE.NKD.AND.ISETA.LT.0) GO TO 30
IH=IH+ISETA
IM=IM+ISETA
IL=IL+ISETA
GO TO 20
C
30 IF (IH.LE.NKD.AND.ISETA.GT.0) GO TO 40
IF (IH.GE.1.AND.ISETA.LT.0) GO TO 40
IH=IH-ISETA
IM=IM-ISETA
IL=IL-ISETA
GO TO 20
C
40 CONTINUE
IF (IM.EQ.IMO) IRECC=1
IF (IM.EQ.IMO) GO TO 60
IRECG = 0
N = IL
DO 45 I=1, NK
IVRWI(I) = N
V(OBWI(I)) = VOBWS(N)

```

```

TOTINT 2
TOTINT 3
TOTINT 4
TOTINT 5
TOTINT 6
DISK2 2
TOTINT 8
TOTINT 9
TOTINT10
TOTINT11
TOTINT12
TOTINT13
TOTINT14
TOTINT15
TOTINT16
TOTINT17
TOTINT18
TOTINT19
TOTINT20
TOTINT21
TOTINT22
TOTINT23
TOTINT24
TOTINT25
TOTINT26
TOTINT27
TOTINT28
TOTINT29
TOTINT30
TOTINT31
TOTINT32
TOTINT33
TOTINT34
TOTINT35
TOTINT36
TOTINT37
TOTINT38
TOTINT39
TOTINT40
TOTINT41
TOTINT42
TOTINT43
TOTINT44
TOTINT45
TOTINT46
TOTINT47
TOTINT48
TOTINT49
TOTINT50
TOTINT51
TOTINT52
TOTINT53

```

```

      LOC=M+(N-1)*IBUMP
      CALL READMS(ND2,DALL(1,1,1),NW,D,LOC)
      IF (KPRCHK.EQ.0) GO TO 45
      CALL PRNT (DALL(1,1,1),NM,NM,NM,2,16HINPUT GEN. AERO ,4)
45  N = N+1SETA
C   60 CONTINUE
C   IF (IRECG.NE.0) GO TO 70
      NWF=2*NTOTAP
      LOC=2*(IG-1)+3
      IF (IND.EQ.3) LOC=LOC+1
C   DC 55 K=1,NK
      N=IVBWI(K)
      LOC=LOC+(N-1)*IBUMP
      CALL READMS(ND2,FALL(1,K),NWF,LOC)
C   55 CONTINUE
C   70 CONTINUE
C   IF (KPRCHK.EQ.0) GO TO 90
      WRITE (NOUT,100) (IVBWI(K),VOBWI(K),K=1,NK)
      LINE$=LINES+3
C   90 CONTINUE
C   INDI=1
      IF (IND.EQ.3.AND.IRECC.NE.0) INDI=0
      CALL INTERP(
1  C,VOBW,DALL,VOBWI,NM,NK,COEF,KPRCHK,IRECG,INDI,RK,C)
      IMO=IM
C   100 FORMAT(1H0,20X, INTERPOLATING FROM THE FOLLOWING IVBW AND VOBW /
1  14 , 10(1X,13,1X,F8.4))
C   RETURN
      END

```

IDIINT54
 IDIINT55
 IDIINT56
 IDIINT57
 IDIINT58
 IDIINT59
 IDIINT60
 IDIINT61
 IDIINT62
 IDIINT63
 IDIINT64
 IDIINT65
 IDIINT66
 IDIINT67
 IDIINT68
 IDIINT69
 IDIINT70
 IDIINT71
 IDIINT72
 IDIINT73
 IDIINT74
 IDIINT75
 IDIINT76
 IDIINT77
 IDIINT78
 IDIINT79
 IDIINT80
 IDIINT81
 IDIINT82
 IDIINT83
 IDIINT84
 IDIINT85
 IDIINT86
 IDIINT87
 IDIINT88
 IDIINT89
 IDIINT90
 IDIINT91
 IDIINT92
 IDIINT93
 IDIINT94

```

SUBROUTINE TFMATF(NTF,NM,TEMP,D,OMEGA,TFC,TMS,NO,MXCRDN,MXCRDD,
1 PHIS,I,PRNT)
C
C COMMON/ZZZ/CASE(48),NIN,NOUT,KROW,LINES
C
C DIMENSION TEMP(2,NM,NM),D(2,NM,NM),TFC(2,NTF,1),TMS(NTF,4)
C DIMENSION PHIS(NTF,1)
C
C DO 500 N=1,NTF
M=TMS(N,4)-NO
DCS=TMS(N,3)
C
C CALL TRFCMX(N,TFC,OMEGA,TFR,TFI,NTF,MXCRDN,MXCRDD)
C
C DO 200 I=1,NM
AR=(D(1,I,M)*TFR-D(2,I,M)*TFI)*DCS
AI=(D(1,I,M)*TFI+D(2,I,M)*TFR)*DCS
C
C DO 200 J=1,NM
TEMP(1,I,J)=AR*PHIS(N,J)
TEMP(2,I,J)=AI*PHIS(N,J)
C
C 200 CONTINUE
C
C IF(I,PRNT,EQ,0)GO TO 250
WRITE(NOUT,1)N,TFR,TFI
1 FORMAT(1H0,20X,DELT AERC FOR ACS TFR,13,5X,TFR=E12.5,5X,
1 TFI=F12.5)
C
C LINES=LINES+2
CALL PRNT(TEMP,NM,NM,2,8HACS AERO,2)
C
C 250 CONTINUE
C
C DO 300 I=1,NM
C DO 300 J=1,NM
D(1,I,J)=D(1,I,J)+TEMP(1,I,J)
D(2,I,J)=D(2,I,J)+TEMP(2,I,J)
C
C 300 CONTINUE
C
C 500 CONTINUE
RETURN
END

```

```

TFMATF 2
TFMATF 3
TFMATF 4
TFMATF 5
TFMATF 6
TFMATF 7
TFMATF 8
TFMATF 9
TFMATF 10
TFMATF 11
TFMATF 12
TFMATF 13
TFMATF 14
TFMATF 15
TFMATF 16
TFMATF 17
TFMATF 18
TFMATF 19
TFMATF 20
TFMATF 21
TFMATF 22
TFMATF 23
TFMATF 24
TFMATF 25
TFMATF 26
TFMATF 27
TFMATF 28
TFMATF 29
TFMATF 30
TFMATF 31
TFMATF 32
TFMATF 33
TFMATF 34
TFMATF 35
TFMATF 36
TFMATF 37
TFMATF 38
TFMATF 39
TFMATF 40
TFMATF 41

```

```

C      SUBROUTINE TFMATM(NTF,NM,TEMP,D,OMEGA,TFC,TMS,NO,MXCRDN,MXORDD,
1      P+IS,EMBAR,NMD,AMULT,I PRNT)
C
C      COMMON/777/CASE(48),NIN,NDUT,KROW,LINES
C
C      DIMENSION TEMP(2,NM,NM),D(2,NM,NM),TFC(2,NTF,1),TMS(NTF,4)
C      DIMENSION PHIS(NTF,1)
C      DIMENSION EMBAR(NMD,NMD)
C
C      DO 500 N=1,NTF
C        M=TMS(N,4)
C        PCS=TMS(N,3)
C
C        CALL TRFCMX(N,TFC,OMEGA,TFR,TFI,NTF,MXCRDN,MXORDD)
C
C        DO 200 I=1,NM
C          II=I+NO
C          AP=EMBAR(II,M)*TFR*AMULT*DCS
C          AI=EMBAR(II,M)*TFI*AMULT*DCS
C          DO 200 J=1,NM
C            TEMP(1,I,J)=AR*PHIS(N,J)
C            TEMP(2,I,J)=AI*PHIS(N,J)
C          CONTINUE
C
C          IF(I PRNT.EQ.0)GO TO 250
C          WRITE(NDUT,1)N,TFR,TFI
C          1  FORMAT(1H0,20X,DELT MASS FOR ACS TFN ,13,5X, TFR= E12.5,5X,
C            TFI= ,E12.5)
C          LINES=LINES+2
C          CALL PRNT(TEMP,NM,NM,2,8HACS MASS,2)
C          CONTINUE
C
C        DO 300 I=1,NM
C          DO 300 J=1,NM
C            D(1,I,J)=D(1,I,J)+TEMP(1,I,J)
C            D(2,I,J)=D(2,I,J)+TEMP(2,I,J)
C          CONTINUE
C
C        500 CONTINUE
C        RETURN
C        END

```

```

TFMATM 2
TFMATM 3
TFMATM 4
TFMATM 5
TFMATM 6
TFMATM 7
TFMATM 8
TFMATM 9
TFMATM10
TFMATM11
TFMATM12
TFMATM13
TFMATM14
TFMATM15
TFMATM16
TFMATM17
TFMATM18
TFMATM19
TFMATM20
TFMATM21
TFMATM22
TFMATM23
TFMATM24
TFMATM25
TFMATM26
TFMATM27
TFMATM28
TFMATM29
TFMATM30
TFMATM31
TFMATM32
TFMATM33
TFMATM34
TFMATM35
TFMATM36
TFMATM37
TFMATM38
TFMATM39
TFMATM40
TFMATM41
TFMATM42
TFMATM43

```

```

C
C
C
SUBROUTINE TRFCMX(N,TFC,OMEGA,TFR,TFI,NTF,MXORDN,MXORDD)
  DIMENSION TFC(2,NTF,1)
  DOUBLE PRECISION AR,BR,AI,BI,POWR,POWI

  AR=0.
  AI=0.
  BR=1.0
  BI=0.
  IF (MXORDN.EQ.0) GO TO 60
  DO 30 I=1,MXORDN,2
    NCRD=I-1
    POWR=OMEGA**NCRD
    PCWI=OMEGA**I
    AR=AR+TFC(1,N,I)*POWR
    AI=AI+TFC(1,N,I+1)*POWI
    CONTINUE
  30
  IF (MXORDD.EQ.0) GO TO 60
  BR=0.
  DO 50 I=1,MXORDD,2
    NCRD=I-1
    POWR=OMEGA**NCRD
    PCWI=OMEGA**I
    BR=BR+TFC(2,N,I)*POWR
    BI=BI+TFC(2,N,I+1)*POWI
    CONTINUE
  50
  CONTINUE
  DEN=BR**2+BI**2
  TFR=(AR*BR+AI*BI)/DEN
  TFI=(AI*BR-AR*BI)/DEN
  RETURN
  END
C

```

```

TRFCMX 3
TRFCMX 4
TRFCMX 5
TRFCMX 6
TRFCMX 7
TRFCMX 8
TRFCMX 9
TRFCMX10
TRFCMX11
TRFCMX12
TRFCMX13
TRFCMX14
TRFCMX15
TRFCMX16
TRFCMX17
TRFCMX18
TRFCMX19
TRFCMX20
TRFCMX21
TRFCMX22
TRFCMX23
TRFCMX24
TRFCMX25
TRFCMX26
TRFCMX27
TRFCMX28
TRFCMX29
TRFCMX30
TRFCMX31
TRFCMX32
TRFCMX33
TRFCMX34
TRFCMX35
TRFCMX36
TRFCMX37
TRFCMX38

```


CFRL0054
CFRL0055
CFRL0056
CFRL0057
CFRL0058
CFRL0059
CFRL0060
CFRL0061
CFRL0062
CFRL0063
CFRL0064
CFRL0065
CFRL0066
CFRL0067
CFRL0068
CFRL0069
CFRL0070
CFRL0071
CFRL0072

```

L62 = L13 + NINTLD*NSBGRP*NAERSB
L63 = L62
LTOT2 = L62 + 2*NINTOTAP

LIOT = MAX0(LIOT1,LIOT2)
WRITE(ROUT,999)LIOT,CORERQ
999 FCRMAT(IHO,20X,IIO,  WORDS CF CORE RQD FOR STEP +++,A10,+++)
CALL FRLOADI
  1 CP,A(L1),A(L2),A(L3),A(L5),A(L6),A(L7),A(L8)
  1 ,A(L9),A(L10),A(L11),A(L12),A(L13),A(L62),A(L63),A(L4),IHD
  1 ,NINTOTAP
  1 ,PHIX,PHIY,PHIZ,A(L3A),NMS,A(L11),A(L2),NACC
  1 ,NINTLD,NM,NSYM,NASYM,NK,NREQ,NABGRP,NSRGRP,NBOX,NSBRETJ
  4 ,NPCXS,NAERSB,NAEROP,A(L51),A(L52)
  5 ,SIGMA,RH00,VTFPS,NG
  6 ,NTFS,NTFA,TFCs,TFCA,TMSS,TMSA,PHISS,PHISA,MXORSN,MXORAN,
  7 MXOPSD,MXORAD,A(L60),A(L61))
  RETURN
END

```

```

SUBROUTINE FGMULT(K,
1 P,NFREQ,VWG,NFLOC,FGL,PINT,NINTLD,NGRP,NFNL,NTOTAP,SCODE)
C
C COMMON/777/HEDP(48),NIN,NCUT,KROW,LINES
C
C COMMON/XTF/NF(100)
C
C GJIVALENCE (NF(20),KPRCHK)
C
C DIMENSION P(2,NINTLD,NFREQ)
C DIMENSION NFNL(3,NGRP)
C DIMENSION PINT(NINTLD,NGRP,1)
C DIMENSION FGL(2,NTOTAP)
C DIMENSION SCODE(1)
C
C DO 150 L=1,NGRP
N=NFNL(2,1)-NFNL(1,1)+1
DO 150 J=1,N
LFC=NFLOC+NFNL(1,1)+J-1
GR=FGL(1,LOC)
GI=FGL(2,LOC)
C
C DF 150 M=1,NINTLD
P(1,M,K)=P(1,M,K)+GR*VWG*PINT(M,L,J)*SCODE(M)
150 P(2,M,K)=P(2,M,K)+GI*VWG*PINT(M,L,J)*SCODE(M)
C
C RETURN
C
C END)

```

```

FGMULT 2
FGMULT 3
FGMULT 4
FGMULT 5
FGMULT 6
FGMULT 7
FGMULT 8
FGMULT 9
FGMULT 10
FGMULT 11
FGMULT 12
FGMULT 13
FGMULT 14
FGMULT 15
FGMULT 16
FGMULT 17
FGMULT 18
FGMULT 19
FGMULT 20
FGMULT 21
FGMULT 22
FGMULT 23
FGMULT 24
FGMULT 25
FGMULT 26
FGMULT 27
FGMULT 28

```


FRL0AD54
 FRL0AD55
 FRL0AD56
 FRL0AD57
 FRL0AD58
 FRL0AD59
 FRL0AD60
 FRL0AD61
 FRL0AD62
 FRL0AD63
 FRL0AD64
 FRL0AD65
 FRL0AD66
 FRL0AD67
 FRL0AD68
 FRL0AD69
 FRL0AD70
 FRL0AD71
 FRL0AD72
 FRL0AD73
 FRL0AD74
 FRL0AD75
 FRL0AD76
 FRL0AD77
 FRL0AD78
 FRL0AD79
 FRL0AD80
 FRL0AD81
 FRL0AD82
 FRL0AD83
 FRL0AD84
 FRL0AD85
 FRL0AD86
 FRL0AD87
 FRL0AD88
 FRL0AD89
 FRL0AD90
 FRL0AD91
 FRL0AD92
 FRL0AD93
 FRL0AD94
 FRL0AD95
 FRL0AD96
 FRL0AD97
 FRL0AD98
 FRL0AD99
 FRL0A100
 FRL0A101
 FRL0A102
 FRL0A103
 FRL0A104
 FRL0A105

```

    IPR11=ITEMPI
    IPR12=ITEMPI
    IPRK1=ITEMPK
    IPRK2=ITEMPK
    IPRK2=ITEMPK-IPRK1*10

    FACTOR=SI7FCT**2

    IHD(5) = NINTLD
    IHD(6) = NACC
    DO 10 I=7,10
    10 IHD(I) = 0
    CALL WRLOAD (1,0,IHD,NOUT,NER)

    READ STALDS INTO PIQ AND WRITE OUT ON LOAD DATASET
    CALL RDUNIT (2,0,PIQ,NOUT,NER)
    CALL WRLOAD (2,0,PIQ,NOUT,NER)

    CALL RDRSP (2,0,0,CR,NOUT,NER)
    CALL WRLOAD (3,0,CR,NOUT,NER)

    CALL RDRSP (3,0,0,OMEGA,NOUT,NER)
    CALL WRLOAD (4,0,OMEGA,NOUT,NER)

    IF (NACC.EQ.0) GO TO 20

    READ (NIN,15) INDACC
    15 FORMAT (6I12)
    CALL HEADNG
    WRITE (NCUT,15) (J, (INDACC(I,J), I=1,2), J=1, NACC)
    16 FORMAT (1H0,5X1H1,2X4HMASS,3X3HDOFF/(1X316) )
    CALL WRLOAD (5,0,INDACC,NOUT,NER)

    20 CONTINUE

    DYNP=SIGMA*PHI0*VTFPS**2*FACTOR/288.
    VWG=DYNP/VTFPS

    DO 300 IG=1,NG
    KPRCHK=0
    KPRCXI=0
    IF (IG.EQ.1) IPR11=0,IG.EQ.1) IPR12) KPRCXI=1
    IF (IG.EQ.1) IPRK1=0,IG.EQ.1) IPRK2) KPRCHK=1
    IF (ITEMPI.GT.99) KPRCXI=1

    IG IS GUST ORIENTATION INDEX

    READ BACK ZERO CHECKS
  
```

FRLOAI06
FRLOAI07
FRLOAI08
FRLOAI09
FRLOAI10
FRLOAI11
FRLOAI12
FRLOAI13
FRLOAI14
FRLOAI15
FRLOAI16
FRLOAI17
FRLOAI18
FRLOAI19
FRLOAI20
FRLOAI21
FRLOAI22
FRLOAI23
FRLOAI24
FRLOAI25
FRLOAI26
FRLOAI27
FRLOAI28
FRLOAI29
FRLOAI30
FRLOAI31
FRLOAI32
FRLOAI33
FRLOAI34
FRLOAI35
FRLOAI36
FRLOAI37
FRLOAI38
FRLOAI39
FRLOAI40
FRLOAI41
FRLOAI42
FRLOAI43
FRLOAI44
FRLOAI45
FRLOAI46
FRLOAI47
FRLOAI48
FRLOAI49
FRLOAI50
FRLOAI51
FRLOAI52
FRLOAI53
FRLOAI54
FRLOAI55
FRLOAI56
FRLOAI57

```

CALL RDRSP(4,IG,1,IHD,ACUT,NER)
IZROS=IHD(4)
I7ROA=IHD(5)

READ IN PIQ FROM NTU
CALL ROUNT (5,0,PIQ,NOUT,NER)
READ IN ALL COEF AND Q FOR THIS IG FROM NTQ

DC 50 I=1,NK
DC 60 J=1,NFREQ
COEF(I,J)=0.0

DC 100 K=1,NFREQ
READ COEF
READ Q
CALL RDRSP(5,IG,K,COEF,NOUT,NER)
CALL RDRSP(6,IG,K,IVBWT,NOUT,NER)
CALL RDRSP(7,IG,K,C(1,1,K),NOUT,NER)
DC 100 J=1,NK
JJ=IVBWT(J)
COEF(JJ,K)=COEF(J)
100 CONTINUE

IF(IPLQ.EQ.0)GC TO 90
IPLQ1=IPLQ/10
IPLQ2=IPLQ-10*IPLQ1
IF(IPLQ1.NE.IG.AND.IPLQ2.NE.IG)GO TO 90
DC 50 J=1,NM
IF(J.LE.NSYM.AND.IZROS.EQ.1)GC TO 50
IF(J.GT.NSYM.AND.IZROA.EQ.1)GC TO 50
DC 40 I=1,NFREQ
PLOT(I)=OMEGA(I)/TWOPI
PLOTV(I)=SQRT(Q(I,J,I)**2+Q(2,J,I)**2)
CALL PLT1(NFREQ,PLOT,PLOTV,1)
WRITE(NOUT,5)IG,J
FORMAT(IH0,20X,Q PLOTS FOR ORIENTATION ,I4, MODE NO. ,I4)
50 CONTINUE
90 CONTINUE

IF(KPRCHK.EQ.0)GO TO 101
CALL HEADNG
WRITE(NOUT,4)IG
FORMAT(IH ,20X, HERE COMES LOTS OF CHECK PRINT FOR ORIENT ,I4)
4
9
WRITE(NOUT,9)DYNP,VWG
FORMAT(IH0,20X,Q = ,F10.4,5X, Q/V = ,F10.4)
CALL PRNT(OMEGA,NFREQ,1,8H OMEGA ,2)
CALL PRNT(COEF,NK,NFREQ,NK,1,8HINT,COEF,2)
CALL PRNT(Q,NM,NFREQ,NM,2,4H Q ,1)

```

FRLOA158
FRLOA159
FRLOA160
FRLOA161
FRLOA162
FRLOA163
FRLOA164
FRLOA165
FRLOA166
FRLOA167
FRLOA168
FRLOA169
FRLOA170
FRLOA171
FRLOA172
FRLOA173
FRLOA174
FRLOA175
FRLOA176
FRLOA177
FRLOA178
FRLOA179
FRLOA180
FRLOA181
FRLOA182
FRLOA183
FRLOA184
FRLOA185
FRLOA186
FRLOA187
FRLOA188
FRLOA189
FRLOA190
FRLOA191
FRLOA192
FRLOA193
FRLOA194
FRLOA195
FRLOA196
FRLOA197
FRLOA198
FRLOA199
FRLOA200
FRLOA201
FRLOA202
FRLOA203
FRLOA204
FRLOA205
FRLOA206
FRLOA207
FRLOA208
FRLOA209

```

101 CALL PRINT(PIQ,NINTLD,NM,NINTLD,1,4H PIQ ,1)
    CONTINUE
    IF (NACC.EQ.0) GO TO 1102
    *** CALC -OMEGA**2 X PHI X Q
    DC 1115 J=1,NFREQ
    W2 = -OMEGA(J)**2/386.088
    CC 1115 I=1,NACC
    AS(1,I,J) = 0.0
    AS(2,I,J) = 0.0
    AA(1,I,J) = 0.0
    AA(2,I,J) = 0.0
    MP = INDACC(1,I)
    L = INDACC(2,I)
    IF (I2R(1,5,1)) GO TO 1109
    DC 1105 K=1,NSYM
    IF (L.EQ.1) PHI = PHIX(MP,K)
    IF (L.EQ.2) PHI = PHIY(MP,K)
    IF (L.EQ.3) PHI = PHIZ(MP,K)
    AS(1,I,J) = AS(1,I,J) + W2*PHI*Q(1,K,J)
    AS(2,I,J) = AS(2,I,J) + W2*PHI*Q(2,K,J)
    1105
    IF (NTFS.EQ.0) GO TO 1109
    DC 1108 N=1,NTFS
    M = TMSS(N,4)
    DCS = TMSS(N,3)
    IF (L.EQ.1) PHI = PHIX(MP,M)
    IF (L.EQ.2) PHI = PHIY(MP,M)
    IF (L.EQ.3) PHI = PHIZ(MP,M)
    IF (PHI.EQ.0.) GO TO 1108
    CALL TRFCMX (N,TFCS,OMEGA(J),TFR,TFI,NTFS,MXORSN,MXORSO)
    AFQ = 0.
    AIO = 0.
    CC 1107 K=1,NSYM
    APQ = ARG+PHISS(N,K)*Q(1,K,J)
    AIO = ARG+PHISS(N,K)*Q(2,K,J)
    AR = (TFR*ARQ-TFI*AIO)*DCS
    AI = (TFR*AIO-TFI*AR)*DCS
    AS(1,I,J) = AS(1,I,J) + W2*PHI*AR
    AS(2,I,J) = AS(2,I,J) + W2*PHI*AI
    1107
    CONTINUE
    1108
    IF (NASYM.EQ.0) GO TO 1114
    IF (I2R(1A,5,1)) GO TO 1114
    DC 1110 KK=1,NASYM

```

[illegible]

FRL0A262
 FRL0A263
 FRL0A264
 FRL0A265
 FRL0A266
 FRL0A267
 FRL0A268
 FRL0A269
 FRL0A270
 FRL0A271
 FRL0A272
 FRL0A273
 FRL0A274
 FRL0A275
 FRL0A276
 FRL0A277
 FRL0A278
 FRL0A279
 FRL0A280
 FRL0A281
 FRL0A282
 FRL0A283
 FRL0A284
 FRL0A285
 FRL0A286
 FRL0A287
 FRL0A288
 FRL0A289
 FRL0A290
 FRL0A291
 FRL0A292
 FRL0A293
 FRL0A294
 FRL0A295
 FRL0A296
 FRL0A297
 FRL0A298
 FRL0A299
 FRL0A300
 FRL0A301
 FRL0A302
 FRL0A303
 FRL0A304
 FRL0A305
 FRL0A306
 FRL0A307
 FRL0A308
 FRL0A309
 FRL0A310
 FRL0A311
 FRL0A312
 FRL0A313

```

105 PS(2,I,J)=0.
    PA(1,I,J)=0.
    PA(2,I,J)=0.
    IF (I7RNS.EQ.1) GO TO 109
    DC 105 K=1, NSYM
    PS(1,I,J)=PS(1,I,J)+PIQ(I,K)*Q(1,K,J)*AMULT
    PS(2,I,J)=PS(2,I,J)+PIQ(I,K)*Q(2,K,J)*AMULT
    IF (NTFS.EQ.0) GO TO 109
    DC 108 N=1, NTFS
    CALL TREC MX(N, TFCS, OMEGA(J), TFR, TFI, NTFS, MXOKSN, MXORD)
    ARQ=0.
    AIQ=0.
    M=TMSS(N,4)
    PCS=TMSS(N,3)
    DC 107 K=1, NSYM
    ARQ=ARQ+PHISS(N,K)*Q(1,K,J)
    AIQ=AIQ+PHISS(N,K)*Q(2,K,J)
    AR=(TFR*ARQ-TFI*AIQ)*DCS
    AI=(TFR*AIQ-TFI*ARQ)*DCS
    PS(1,I,J)=PS(1,I,J)+PIQ(I,M)*AR*AMULT
    PS(2,I,J)=PS(2,I,J)+PIQ(I,M)*AI*AMULT
103 CONTINUE
109 CONTINUE
    IF (NASYM.EQ.0) GO TO 114
    IF (I2RDA.EQ.1) GO TO 114
    DC 110 K=1, NASYM
    KK=K+NSYM
    PA(1,I,J)=PA(1,I,J)+PIQ(I,KK)*Q(1,KK,J)*AMULT
    PA(2,I,J)=PA(2,I,J)+PIQ(I,KK)*Q(2,KK,J)*AMULT
    IF (NTFA.EQ.0) GO TO 114
    DC 113 N=1, NTFA
    CALL TREC MX(N, TFCA, OMEGA(J), TFR, TFI, NTFA, MXORAN, MXORD)
    ARQ=0.
    AIQ=0.
    M=TMCA(N,4)
    PCS=TMCA(N,3)
    DC 112 K=1, NASYM
    KK=K+NSYM
    ARQ=ARQ+PHISSA(N,K)*Q(1,KK,J)
    AIQ=AIQ+PHISSA(N,K)*Q(2,KK,J)
    AR=(TFR*ARQ-TFI*AIQ)*DCS
    AI=(TFR*AIQ-TFI*ARQ)*DCS
    PS(1,I,J)=PS(1,I,J)+PIQ(I,M)*AR*AMULT
    PS(2,I,J)=PS(2,I,J)+PIQ(I,M)*AI*AMULT
113 CONTINUE
114 CONTINUE
115 CONTINUE
    IF (KPRCHK.EQ.0) GO TO 102
    IF (I7RNS.EQ.0) CALL PRINT
    1 PS, NINTID, NPEEQ, NINTID, 2, 20H SYM INERTIAL LOADS , 5)
    IF (NASYM.NE.0) AND. I2RDA.EQ.0) CALL PRINT
  
```

```

1 PA,NINTLD,NFREQ,NINTLD,2,20HASYM INERTIAL LOADS ,5)
102 CONTINUE
C
C
C CALC DYNP X MOTION DEPENDENT AERO FORCES X Q
C
C DO 200 I=1,NK
C
C READ IN PAQS(L) AND PAQA(L) FOR THIS IG
C
C IF (I7ROS.EQ.1) GO TO 116
C
C CALL RDUNIT (7,L,PAQS,NCUT,NER)
C
C 116 CONTINUE
C
C IF (NASYM.EQ.0) GO TO 117
C IF (I7ROA.EQ.1) GO TO 117
C
C CALL RDUNIT (8,L,PAQA,NCUT,NER)
C
C 117 CONTINUE
C IF (KPRCHK.EQ.0) GO TO 118
C IF (I7ROS.EQ.0) CALL PRNT(
C 1 PAQS,NINTLD,NSYM,NINTLD,2,4HPAQS ,1)
C IF (NASYM.NE.0 .AND. I7ROA.EQ.0) CALL PRNT(
C 1 PAQA,NINTLD,NASYM,NINTLD,2,4HPAQA ,1)
C
C 118 CONTINUE
C DO 140 J=1,NFREQ
C AMULT=DYNP*COEF(L,J)
C DO 140 I=1,NINTLD
C
C IF (I7ROS.EQ.1) GO TO 125
C DO 120 K=1,NSYM
C PS(I,I,J)=PS(I,I,J)+
C 1 (PAQS(I,I,K)*Q(I,K,J)-PAQS(2,I,K)*Q(2,K,J))*AMULT
C 120 PS(I,I,J)=PS(I,I,J)+
C 1 (PAQS(I,I,K)*Q(2,K,J)+PAQS(2,I,K)*Q(1,K,J))*AMULT
C 1 IF (NTFS.EQ.0) GO TO 125
C DO 124 N=1,NTFS
C CALL TRFCMX(N,TFCS,OMEGA(J),TFR,TFI,NTFS,MXORSN,MXORSO)
C ARQ=0.
C AIQ=0.
C M=TMSS(N,4)
C NCSS=TMSS(N,3)
C DO 123 K=1,NSYM
C APQ=ARQ+PHISS(N,K)*Q(1,K,J)
C 123 AIQ=AIQ+PHISS(N,K)*Q(2,K,J)
C AP=(TFR*ARQ-TFI*AIQ)*DCS
C AI=(TFR*AIQ-TFI*ARQ)*DCS
C PS(I,I,J)=PS(I,I,J)+(PAQS(1,I,J)+PAQS(2,I,J)*AI)*AMULT

```

FRL0A314
 FRL0A315
 FRL0A317
 FRL0A318
 FRL0A319
 FRL0A320
 FRL0A321
 FRL0A322
 FRL0A323
 FRL0A324
 FRL0A325
 FRL0A326
 FRL0A327
 FRL0A328
 FRL0A329
 FRL0A330
 FRL0A331
 FRL0A332
 FRL0A333
 FRL0A334
 FRL0A335
 FRL0A336
 FRL0A337
 FRL0A338
 FRL0A339
 FRL0A340
 FRL0A341
 FRL0A342
 FRL0A343
 FRL0A344
 FRL0A345
 FRL0A346
 FRL0A347
 FRL0A348
 FRL0A349
 FRL0A350
 FRL0A351
 FRL0A352
 FRL0A353
 FRL0A354
 FRL0A355
 FRL0A356
 FRL0A357
 FRL0A358
 FRL0A359
 FRL0A360
 FRL0A361
 FRL0A362
 FRL0A363
 FRL0A364
 FRL0A365

319

```

C      READ IN NFNL SB,PINTZ
C      CALL ROUNIT(13,0,NFNL SB,NOUT,NER)
C      CALL ROUNIT(14,0,PINTZ ,NOUT,NER)
C      READ IN PINTY
C      CALL ROUNIT(15,0,PINTY ,NOUT,NER)
C
C      DO 260 J=1,NFREQ
C      IF (IZROS.FQ.1) GO TO 401
C
C      CALL RDRSP(8,IG,J,FGPS,NOUT,NER)
C      NFLOC=0
C
C      NEW CALC SYMMETRIC INTGD GUST LOADS
C
C      CALL EGMULT(J,
C      1 PS,NFREQ,VWG,NFLOC,FGPS,PINTP,NINTLD,NABGRP,NFNL AB,NTOTAP,
C      2 SYMCD)
C      NFLOC=NB CX
C
C      CALL EGMULT(J,
C      1 PS,NFREQ,VWG,NFLOC,FGPS,PINTZ,NINTLD,NSBGRP,NFNL SB,NTOTAP,
C      2 SYMCD)
C      NFLOC=NB CX+NSRETO
C
C      CALL EGMULT(J,
C      1 PS,NFREQ,VWG,NFLOC,FGPS,PINTY,NINTLD,NSBGRP,NFNL SB,NTOTAP,
C      2 SYMCD)
C      401 IF (NASYM.FQ.0) GO TO 402
C      IF (IZROA.EQ.1) GO TO 402
C      CALL RDRSP(9,IG,J,FGPA,NOUT,NER)
C      NFLOC=0
C
C      NEW CALC ANTISYMMETRIC INTGD GUST LOADS
C
C      CALL EGMULT(J,
C      1 PA,NFREQ,VWG,NFLOC,FGPA,PINTP,NINTLD,NABGRP,NFNL AB,NTOTAP,
C      2 ASYCD)
C      NFLOC=NB CX
C
C      CALL EGMULT(J,
C      1 PA,NFREQ,VWG,NFLOC,FGPA,PINTZ,NINTLD,NSBGRP,NFNL SB,NTOTAP,
C      2 ASYCD)
C      NFLOC=NB CX+NSRETO

```

```

FRLOA418
FRLOA419
FRLOA420
FRLOA421
FRLOA422
FRLOA423
FRLOA424
FRLOA425
FRLOA426
FRLOA427
FRLOA428
FRLOA429
FRLOA430
FRLOA431
FRLOA432
FRLOA433
FRLOA434
FRLOA435
FRLOA436
FRLOA437
FRLOA438
FRLOA439
FRLOA440
FRLOA441
FRLOA442
FRLOA443
FRLOA444
FRLOA445
FRLOA446
FRLOA447
FRLOA448
FRLOA449
FRLOA450
FRLOA451
FRLOA452
FRLOA453
FRLOA454
FRLOA455
FRLOA456
FRLOA457
FRLOA458
FRLOA459
FRLOA460
FRLOA461
FRLOA462
FRLOA463
FRLOA464
FRLOA465
FRLOA466
FRLOA467
FRLOA468
FRLOA469

```

FRLOA470
 FRLOA471
 FRLOA472
 FRLOA473
 FRLOA474
 FRLOA475
 FRLOA476
 FRLOA477
 FRLOA478
 FRLOA479
 FRLOA480
 FRLOA481
 FRLOA482
 FRLOA483
 FRLOA484
 FRLOA485
 FRLOA486
 FRLOA487
 FRLOA488
 FRLOA489
 FRLOA490
 FRLOA491
 FRLOA492
 FRLOA493
 FRLOA494
 FRLOA495
 FRLOA496
 FRLOA497
 FRLOA498
 FRLOA499
 FRLOA500
 FRLOA501
 FRLOA502
 FRLOA503
 FRLOA504
 FRLOA505
 FRLOA506
 FRLOA507
 FRLOA508
 FRLOA509
 FRLOA510
 FRLOA511
 FRLOA512
 FRLOA513
 FRLOA514
 FRLOA515
 FRLOA516
 FRLOA517
 FRLOA518
 FRLOA519
 FRLOA520
 FRLOA521

```

C      CALL FGMLT(J,
1      PA,NFREQ,VWG,NFLUC,FGPA,PINTY,NINTLD,NSBGRP,NFNL SB,NTOTAP,
2      ASMCOD)
C
C      402 CONTINUE
C      260 CONTINUE
C      PRINT LOADS IF DESIRED
C      IF(KPRCHK.EQ.0)GO TO 270
C      IF(I7ROS.EQ.0)CALL PRNT(
1      PS,NINTLD,NFREQ,NINTLD,2,16H SYM INTGD LOADS ,4)
C      IF(NASYM.NE.0.AND.IZROA.EQ.0)CALL PRNT(
1      PA,NINTLD,NFREQ,NINTLD,2,16HASYM INTGD LOADS ,4)
C      270 CONTINUE
C
C      IF(KPRCXI.EQ.0)GO TO 280
C      IF(I7ROS.EQ.1)GO TO 290
C      CALL HEADNG
C      WRITE(UNIT,2)IG
C      FORMAT(1H ,20X, SYM. UNIT GUST LOADS FOR ORIENTATION NO. ,I4)
2      CALL INTLPR(PS,NINTLD,NFREQ,OMEGA)
290 CONTINUE
C      IF(NASYM.EQ.0)GO TO 280
C      IF(I7ROA.EQ.1)GO TO 280
C      CALL HEADNG
C      WRITE(UNIT,3)IG
C      FORMAT(1H ,20X, ASYM. UNIT GUST LOADS FOR ORIENTATION NO. ,I4)
3      CALL INTLPR(PA,NINTLD,NFREQ,OMEGA)
C      280 CONTINUE
C
C      NOW PLOT SELECTED INTGD LOAD
C
C      IF(IPLL.EQ.0)GO TO 299
C      IF(IPLL.GT.NINTLD)GO TO 299
C      IF(I7ROS.EQ.1)GO TO 298
C
C      DC 295 I=1,NFREQ
C      PLOT(I)=OMEGA(I)/TWCP1
C      PLOTV(I)=SQRT(PS(1,IPLL,I)**2+PS(2,IPLL,I)**2)
295 CALL PLT1(NFREQ,PLOT,PLOTV,1)
C      WRITE(UNIT,6)IG,IPLL
C      FORMAT(1H ,20X, SYM UNIT GUST LOADS FOR ORIENT ,I4, LOAD ,I4)
6      CONTINUE
298 IF(NASYM.EQ.0)GO TO 299
C      IF(I7ROA.EQ.1)GO TO 299
  
```

FRL0A522
FRL0A523
FRL0A524
FRL0A525
FRL0A526
FRL0A527
FRL0A528
FRL0A529
FRL0A530
FRL0A531
FRL0A532
FRL0A533
FRL0A534
FRL0A535
FRL0A536
FRL0A537
FRL0A538
FRL0A539
FRL0A540
FRL0A541
FRL0A542
FRL0A543
FRL0A544
FRL0A545
FRL0A546
FRL0A547
FRL0A548
FRL0A549
FRL0A550
FRL0A551
FRL0A552
FRL0A553
FRL0A554
FRL0A555
FRL0A556
FRL0A557
FRL0A558

```

SUBROUTINE INTLPR(F, NP, NF, OMEGA)
COMMON/777/HEDR(48), NIN, NOUT, ICK, IC, IPRNT, NER
DIMENSION F(2, NP, NF)
DIMENSION FMO(8)
DIMENSION OMEGA(NF)
DATA TWOP1/6.283185/
FORMAT(1H0, *FREQ HZ*, 8(4X, I4, 7X))
FORMAT(1H0, 30X, *INT: LOADS - MODULUS*/ 1H0, 30X, *INT. LOAD NO.**)
FORMAT(1H, *F7.3, 8F15.7)
IS=1
IF=8
IF (IF.GT.NP) IF=NP
IF (IC.LT.ICK) GO TO 401
CALL HEADNG
CONTINUE
WRITE(NOUT, 9)
JF=IF-IS+1
WRITE(NOUT, 7) (J, J=IS, IF)
IC=IC+6
DC 500 I=1, NF
FREQ=OMEGA(I)/TWOP1
IC=IC+1
IF (IC.LE.ICK) GO TO 692
CALL HEADNG
WRITE(NOUT, 9)
WRITE(NOUT, 7) (J, J=IS, IF)
IC=IC+6
CONTINUE
DC 490 J=1, JF
JJ=J+IS-1
FMO(J)=SQRT(F(1, JJ, I)**2+F(2, JJ, I)**2)
WRITE(NOUT, 8) FREQ, (FMO(J), J=1, JF)
IF (IF.GE.NP) GO TO 510
IS=IF+1
IF=IF+8
GO TO 400
CONTINUE
RETURN
END

```

```

INTLPR 2
INTLPR 3
INTLPR 4
INTLPR 5
INTLPR 6
INTLPR 7
INTLPR 8
INTLPR 9
INTLPR10
INTLPR11
INTLPR12
INTLPR13
INTLPR14
INTLPR15
INTLPR16
INTLPR17
INTLPR18
INTLPR19
INTLPR20
INTLPR21
INTLPR22
INTLPR23
INTLPR24
INTLPR25
INTLPR26
INTLPR27
INTLPR28
INTLPR29
INTLPR30
INTLPR31
INTLPR32
INTLPR33
INTLPR34
INTLPR35
INTLPR36
INTLPR37
INTLPR38
INTLPR39
INTLPR40

```



```

2 2F15.6, 10X, 7HM-Y/Q = , 2F15.6 / , )
100 FFORMAT ( //25X, 26HTOTALS ON LIFTING SURFACES )
110 FFORMAT ( //10X, 7HCZ =, 2F15.6, 10X, 7HCY =, 2F15.6 / 10X,
1 7HCM =, 2F15.6, 10X, 7HCN =, 2F15.6 / 10X,
2 7HCSL =, 2F15.6 / , )
120 FFORMAT ( //25X, 27HTOTALS ON ENTIPE AIRCRAFT )
130 FFORMAT ( //15, F12.4, 3X, 4F14.5 )
140 FFORMAT ( //1H1 )
150 FFORMAT ( //48H STATION SPANWISE SPAN LOAD (C*CL)/(C-BAR)
1 WRITE (NOUT,10) KR COORD. /IMAG. /)

C
WRITE (NOUT,20)
12 = 0
LP = 1
LPAGE = 1
DO 170 J = 1, NSTRIP
11 = 12 + 1
12 = 12 + NCARRAY(LP)
DO 160 I = 1, 12
XFC = (XIC(I)-XIJ(J)) / CS(J)
IF (YS(J).EQ.0.0) DCP(I) = DCP(I)*2.0
DCP(I) = DCP(I)/DELA(I)
WRITE (NOUT,70) LP, J, I, XFC, XIC(I), YS(J), ZS(J), DCP(I)
IF (LPAGE*LNES.NE. I) GO TO 160
LPAGE = LPAGE + 1
WRITE (NOUT,20)
160 CONTINUE
IF (I2.E..NCARRAY(LP)) LP=LP+1
170 CONTINUE

C
C
C
***
DELT = 1.0
IF (INDSYM.EQ.0) DELT=-1.0
SYMA = 1.0+DELT
SYMB = 1.0-DELT
DO 190 I=1,NSTRIP
CPO(I) = 0.0
CPI(I) = 0.0
190 CONTINUE
WRITE (NOUT,30)
IF (NA.NE.0) WRITE(NOUT,40)
WRITE (NOUT,50)
12 = 0
LP = 1
DO 200 J=1,NSTRIP
11 = 12+1
12 = 12+NCARRAY(LP)
IF (I2.FO.NCARRAY(LP)) LP=LP+1
Y114S = 0.25*(C(J) + XIJ(J)

```

AER02 50
AER02 51
AER02 52
AER02 53
AER02 54
AER02 55
AER02 56
AER02 57
AER02 58
AER02 59
AER02 60
AER02 61
AER02 62
AER02 63
AER02 64
AER02 65
AER02 66
AER02 67
AER02 68
AER02 69
AER02 70
AER02 71
AER02 72
AER02 73
AER02 74
AER02 75
AER02 76
AER02 77
AER02 78
AER02 79
AER02 80
AER02 81
AER02 82
AER02 83
AER02 84
AER02 85
AER02 86
AER02 87
AER02 88
AER02 89
AER02 90
AER02 91
AER02 92
AER02 93
AER02 94
AER02 95
AER02 96
AER02 97
AER02 98
AER02 99
AER02 100
AER02 101

```

CN(J) = (0.0,0.0)
CPI(J) = (0.0,0.0)
SPLD(J) = (0.0,0.0)
DC 220 I=1,12
CN(J) = CN(J) + DCP(I)*DELX(I)
CM(J) = CM(J) + DCP(I)*DELX(I)*(XIC(I) - XI14S)
CONTINUE
YCS = VS(J)/REFS
CH2 = CS(J)*2
JL = ISTR(J)
SPLD(JL) = SPLD(JL) + CN(J) / REFC
CM(J) = CM(J)/CS(J)
CM(J) = -CM(J)/CH2
ARSCNJ = SQRT((REAL(CN(J))**2 + (AIMAG(CN(J))**2)
IF (ARSCNJ .LE. 0.000001) GO TO 230
CPR(J) = -REAL(CM(J))/REAL(CN(J)) + 0.25
IF (KR .LE. 0.0001) GO TO 240
CPI(J) = -AIMAG(CM(J))/AIMAG(CN(J)) + 0.25
GO TO 250
230 CONTINUE
CPR(J) = 0.0
240 CONTINUE
CPI(J) = 0.0
250 CONTINUE
WRITE (NOUT, 72) J,YS(J), ZS(J), YOS,CN(J),CM(J),CPR(J),CPI(J)
CONTINUE
WRITE (NOUT,140)
WRITE (NOUT,150)
DC 270 JL = 1, MAXSTR
WRITE (NOUT,130) JL, COORD(JL), SPLD(JL)
CONTINUE
IF (NB.EG.0) GO TO 300
L2 = 0
DC 290 N=1, NB
CZB(N) = (0.0,0.0)
CYB(N) = (0.0,0.0)
CMB(N) = (0.0,0.0)
CAR(N) = (0.0,0.0)
L1 = L2+1
L2 = L2 + NSREA(N)
SBL = L2 + XIS2(L2) - XIS1(L1)
WRITE (NOUT,60)
DC 280 LH=L1,L2
IF (YB(LB).EQ.0.0) FZ(LB) = FZ(LB)*2.0
IF (YB(LB).EQ.0.0) FY(LB) = FY(LB)*2.0
LX = LX+1
XSR = 0.5*(XIS1(LB) + XIS2(LB))
DXSR = XIS2(LB) - XIS1(LB)
XCL = (XSR - XIS1(L1))/ SBL
FZLA = FZ(LB)/ DXSB
FYLB = FY(LB)/ DXSB
WRITE (NOUT, 72) LB, YB(N), ZR(N), XOL, FZLB, FYLB

```

AER02102
AER02103
AER02104
AER02105
AER02106
AER02107
AER02108
AER02109
AER02110
AER02111
AER02112
AER02113
AER02114
AER02115
AER02116
AER02117
AER02118
AER02119
AER02120
AER02121
AER02122
AER02123
AER02124
AER02125
AER02126
AER02127
AER02128
AER02129
AER02130
AER02131
AER02132
AER02133
AER02134
AER02135
AER02136
AER02137
AER02138
AER02139
AER02140
AER02141
AER02142
AER02143
AER02144
AER02145
AER02146
AER02147
AER02148
AER02149
AER02150
AER02151
AER02152
AER02153

```

CZB(N) = CZB(N) + FZ(LR)
CYB(N) = CYB(N) + FY(LR)
CMR(N) = CMR(N) - FZ(LR) * (XSB-XIS1(L1))
CNR(N) = CNR(N) - FY(LR) * (XSB-XIS1(L1))
280 CONTINUE
CZB(N) = CZB(N)/REFA
CYB(N) = CYB(N)/REFA
CMR(N) = CMR(N)/(REFA*REFC)
CNR(N) = CNR(N)/(REFA*REFC)
WRITE (NOUT, 90) N, CZB(N), CMR(N), CYB(N), CNR(N)
290 CONTINUE
300 CONTINUE
WRITE (NOUT, 140)
CZT = (0.0, 0.0)
CYT = (0.0, 0.0)
CMT = (0.0, 0.0)
CAT = (0.0, 0.0)
CLT = (0.0, 0.0)
320 J=1, NSTRI
CH2 = CS(J)*2
XI14S = 0.25*CS(J) + XIJ(J)
CMULT = 2.0*FE(J)*CG(J)
SMULT = 2.0*FF(J)*SG(J)
GUCJ = 1.0
IF (ABS(YS(J)).LE.0.0001 .AND. .ARS(CG(J)).LE.0.0001) GUCJ=0.5
CXT = CH2*CM(J) - CS(J)*CN(J)*CMULT * GUCJ
CZT = CZT + CS(J)*CN(J)*CMULT * GUCJ
CYT = CYT - CS(J)*CN(J)*SMULT * GUCJ
CMT = CMT + CXT * CMULT * GUCJ
CNT = CNT - CXT * SMULT * GUCJ
CLT = CLT + CS(J)*CN(J)*YS(J)*CG(J) + 7S(J)*SG(J)*2.0*EE(J)*GUCJ
320 CONTINUE
WRITE (NOUT, 100)
CZT = CZT/REFA
CZTS = SYMA * CZT
CYT = CYT/PEFA
CYTS = SYMB * CYT
CMT = CMT/(REFA*REFC)
CNT = CNT/(RHEFA*PEFC)
CLTS = SYMA * CNT
CLTS = SYMB * CLT / (2.0*REFA*REFC)
WRITE (NOUT, 110) CZTS, CYTS, CMTS, CNTS, CLTS
IF (N.B.EQ.0) GO TO 380
L1 = 0
L2 = 0
J1 = 1
J2 = 1
330 J=J1, J2
GLCH = 1.0
IF (ABS(YR(J)).LE.0.0001) GLCH = 0.5
L1 = L2+1

```

AER02154
AER02155
AER02156
AER02157
AER02158
AER02159
AER02160
AER02161
AER02162
AER02163
AER02164
AER02165
AER02166
AER02167
AER02168
AER02169
AER02170
AER02171
AER02172
AER02173
AER02174
AER02175
AER02176
AER02177
AER02178
AER02179
AER02180
AER02181
AER02182
AER02183
AER02184
AER02185
AER02186
AER02187
AER02188
AER02189
AER02190
AER02191
AER02192
AER02193
AER02194
AER02195
AER02196
AER02197
AER02198
AER02199
AER02200
AER02201
AER02202
AER02203
AER02204
AER02205

```

      L2 = L2+NSBEA(J)
      CZT = CZT + (CZB(J) - GLCB)
      CMT = (CMB(J) - CZB(J)) * (XIS1(L1) - XM) / REFC) * GLCB
      CLT = CLT + REFA * CZB(J) * YR(J) * GLCB
330 CONTINUE
      L1 = 0
      L2 = 0
      J1 = NH-NBY+1
      J2 = NR
      J1M1 = J1-1
      IF (J1M1.EQ.0) GC TC 360
      DC 350 JX=1, J1M1
      L2 = L2+NSBEA(JX)
350 CONTINUE
360 CONTINUE
      DC 370 J=J1, J2
      GLCB = 1.0
      IF (ABS(YR(J)) .IF. 0.0001) GLCB = 0.5
      L1 = L2+1
      L2 = L2+NSBEA(J)
      CYT = CYT + CYB(J) * GLCB
      CNT = CNT + (CMB(J) - CYB(J)) * (XIS1(L1) - XM) / REFC) * GLCB
      CLT = CLT - REFA * CYB(J) * ZR(J) * GLCB
370 CONTINUE
380 CONTINUE
      CZT = SYMA * CZT
      CYT = SYMB * CYT
      CMT = SYMA * CMT
      CNT = SYMB * CNT
      CLT = SYMB * CLT / (2.0*REFS*REFFA)
      WRITE (NCUT,120) CZT, CYT, CMT, CNT, CLT
      WRITE (NCUT,110)
      RETURN
      END

```

AER02206
 AER02207
 AER02208
 AER02209
 AER02210
 AER02211
 AER02212
 AER02213
 AER02214
 AER02215
 AER02216
 AER02217
 AER02218
 AER02219
 AER02220
 AER02221
 AER02222
 AER02223
 AER02224
 AER02225
 AER02226
 AER02227
 AER02228
 AER02229
 AER02230
 AER02231
 AER02232
 AER02233
 AER02234
 AER02235
 AER02236
 AER02237
 AER02238
 AER02239

```

C
SUBROUTINE CTIRM(AMCDNO,EMBAR,CAY,PS,PA,THRUST)
  DIMENSION THRUST(1)
  DIMENSION AMODNO(20), EMBAR(1), CAY(1)
  COMMON NAA,A(1)
  COMMON /XTE/NF(100)
  EQUIVALENCE (NF(6),NSYM), (NF(7),NASYM), (NF(8),NMODES)
  1 , (NF(10),NENG5), (NF(32),NINTLD)
  2 , (NF(43),NBOX), (NF(44),NSBE)
  3 , THRUST,IVBW

C
COMMON /ZZZ/HFDR(48),NIN,NOUT,KROW,LINES,IPRINT,NER

C
DATA CORFRQ/8H TPIM /

C
NMS = 2
IF (AMODNO(4).NE.0) NMS = NMS+AMODNO(5)-AMODNO(4)+1
NMA = 3
IF (AMODNO(14).NE.0) NMA = NMA+AMODNO(15)-AMODNO(14)+1

C
11 = NAA
L2 = L1 + NMODES
L14 = L2 + NMCDES
L15 = L14 + NINTLD*NENG5
L3 = L15 + NSYM*NENG5
L4 = L3 + NMS*NMS
L5 = L4 + NMS
LSYM = L5 + 2*NSYM*NSYM
L6 = L3
L7 = L6 + NMA*NMA
L8 = L7 + NMA
LASM = L8 + 2*NASYM*NASYM
L11 = L3
LPIQ = L11 + NINTLD*NMODES
L12 = L3
L13 = L12 + 2*NINTLD*NSYM
LF = L13 + 2*NINTLD*NASYM

C
LMAX = MAX0(LSYM,LASM,LF,LPIQ)

C
WRITE(INOUT,999) LMAX,CORFRQ
999 FORMAT(1H0,20X,I10, ' WORDS OF CORE REQD FOR STEP +++,A10,+++)

C
CALL TPIM (AMCDNO,EMBAR,CAY,NSYM,NASYM,NMODES,NINTLD,NMS,NMA
  1 , A(L1),A(L2),A(L3),A(L4),A(L5),A(L6),A(L7),A(L8)
  2 , PS,PA,A(L11),A(L12),A(L13),A(L14),A(L15)
  3 , THRUST,IVBW)

C
L2 = L1 + NMODES
L3 = L2 + 2*NBOX
L4 = L3 + 2*NSHE
L5 = L4 + 2*NSBE
L6 = L5 + 2*MAX0(NBOX,NSBE)*MAX0(NSYM,NASYM)

```

CTIRM 2
 CTIRM 3
 CTIRM 4
 CTIRM 5
 CTIRM 6
 CTIRM 7
 CTIRM 8
 CTIRM 9
 CTIRM 10
 CTIRM 11
 CTIRM 12
 CTIRM 13
 CTIRM 14
 CTIRM 15
 CTIRM 16
 CTIRM 17
 CTIRM 18
 CTIRM 19
 CTIRM 20
 CTIRM 21
 CTIRM 22
 CTIRM 23
 CTIRM 24
 CTIRM 25
 CTIRM 26
 CTIRM 27
 CTIRM 28
 CTIRM 29
 CTIRM 30
 CTIRM 31
 CTIRM 32
 CTIRM 33
 CTIRM 34
 CTIRM 35
 CTIRM 36
 CTIRM 37
 CTIRM 38
 CTIRM 39
 CTIRM 40
 CTIRM 41
 CTIRM 42
 CTIRM 43
 CTIRM 44
 CTIRM 45
 CTIRM 46
 CTIRM 47
 CTIRM 48
 CTIRM 49
 CTIRM 50
 CTIRM 51
 CTIRM 52
 CTIRM 53

CTRIM 54
CTRIM 55
CTRIM 56
CTRIM 57
CTRIM 58
CTRIM 59

C CALL SPANLO (NQUT,NFR,IVBW,NBOX,NSBE
1 , A(L1),A(L2),A(L3),A(L4),A(L5),A(L6))
C RETURN
END

```

SUBROUTINE SPANLD (NCUT,NER,IVPW,NBOX,NSBE,0,DCP,FZ,FY,D,DB,A)
  DIMENSION Q(1), A(1)
  COMPLEX DCP(1), FZ(1), FY(1)
  COMPLEX Q(NBOX,1), DB(NSBE,1)
  COMMON /XTE/NF(100)
  EQUIVALENCE (NF(6),NSYM), (NF(7),NASYM)
  COMMON /ARCCOM/ NTI, MODES
  X = NP, MSTRIP, NSMAX, NCMAX, NTOTAL, NB, MSBE, MBE
  Y = ND, NE, NRY, NRZ, NTO, NTP, NTY, NTZ
  I = NIVS, NIZS, MAXGR, MAXSTR, NSBEIC, NSTRIP, KP, XM, REFA, REFC
  J = REFS, FMACH, LINES
  DIMENSION IHD(50),RHD(50)
  EQUIVALENCE (IHD(1),RHD(1))
  CALL RDAERO (4HHEAD,0,IHD,NCUT,NER)
  NP = IHD(6)
  NRY = NR
  NRZ = NR
  NSTRIP = IHD(9)
  MAXSTR = IHD(10)
  NP = IHD(11)
  FMACH = RHD(21)
  REFA = RHD(22)
  REFS = RHD(23)
  REFC = RHD(24)
  XM = RHD(25)
  CALL RDAERO (4HGEOM,0,A,NCUT,NER)
  L1 = L1
  L2 = L1 + NBOX
  L3 = L2 + NBOX
  L4 = L3 + NBOX
  L5 = L4 + NSTRIP
  L6 = L5 + NSTRIP
  L7 = L6 + NSTRIP
  L8 = L7 + NSTRIP
  L9 = L8 + NSTRIP
  L10 = L9 + NSTRIP
  L11 = L10 + NSTRIP
  L12 = L11 + NSTRIP
  L13 = L12 + MAXSTR
  L14 = L13 + NP
  L15 = L14 + NP
  L16 = L15 + NP
  L17 = L16 + NP
  SPANLD 2
  SPANLD 3
  SPANLD 4
  SPANLD 5
  SPANLD 6
  SPANLD 7
  SPANLD 8
  SPANLD 9
  SPANLD 10
  SPANLD 11
  ARCCOM 2
  ARCCOM 3
  ARCCOM 4
  ARCCOM 5
  ARCCOM 6
  SPANLD 13
  SPANLD 14
  SPANLD 15
  SPANLD 16
  SPANLD 17
  SPANLD 18
  SPANLD 19
  SPANLD 20
  SPANLD 21
  SPANLD 22
  SPANLD 23
  SPANLD 24
  SPANLD 25
  SPANLD 26
  SPANLD 27
  SPANLD 28
  SPANLD 29
  SPANLD 30
  SPANLD 31
  SPANLD 32
  SPANLD 33
  SPANLD 34
  SPANLD 35
  SPANLD 36
  SPANLD 37
  SPANLD 38
  SPANLD 39
  SPANLD 40
  SPANLD 41
  SPANLD 42
  SPANLD 43
  SPANLD 44
  SPANLD 45
  SPANLD 46
  SPANLD 47
  SPANLD 48
  SPANLD 49

```

```

C      L18 = L17 + NR
C      CALL HFDNG
C
C      CALL RDAERO (4HGEO5,0,A(L18),NOUT,NER)
C
C      L19 = L18 + NSBE
C      N = L19-1
C      M = N+2*NSRE
C      DC 50 I=1,NSBE
C      N = N+1
C      M = M+1
C      50 A(N) = A(M)
C
C      CALL RDAERO (4HDPSP,IVBW,D,NOUT,NER)
C
C      DC 100 I=1,NBOX
C      DCP(I) = 0.0
C      DC 100 J=1,NSYM
C      100 DCP(I) = DCP(I) + D(I,J)*Q(J)
C
C      IF (NSBE.EQ.0) GO TO 150
C
C      CALL RDAERO (4HDZSP,IVBW,DB,NOUT,NER)
C
C      DC 110 I=1,NSRE
C      F7(I) = 0.0
C      DC 110 J=1,NSYM
C      110 F7(I) = F7(I) + DB(I,J)*Q(J)
C
C      CALL RDAERO (4HDYSP,IVBW,DB,NOUT,NER)
C
C      DC 120 I=1,NSBE
C      FY(I) = 0.0
C      DC 120 J=1,NSYM
C      120 FY(I) = FY(I) + DB(I,J)*Q(J)
C
C      150 L20 = L19 + NSBE
C      L21 = L20 + 2*NSSTRIP
C      L22 = L21 + 2*NSSTRIP
C      L23 = L22 + 2*NSSTRIP
C      L24 = L23 + 2*NSSTRIP
C      L25 = L24 + 2*NSSTRIP
C      L26 = L25 + 2*NSSTRIP
C      L27 = L26 + 2*NSSTRIP
C      L28 = L27 + 2*NSSTRIP
C
C      CALL AERO2 (NCUT,INDSYM,DCP,F7,FY
C      1 , A(L1),A(L2),A(L3),A(L4),A(L5),A(L6),A(L7),A(L8),A(L9),A(L10)
C      2 , A(L11),A(L12),A(L13),A(L14),A(L15),A(L16),A(L17),A(L18),A(L19)
C      3 , A(L20),A(L21),A(L22),A(L23),A(L24),A(L25),A(L26),A(L27),A(L28))
C
C      IF (INDSYM.EQ.1) GO TO 200

```

```

SPANLD50
SPANLD51
SPANLD52
SPANLD53
SPANLD54
SPANLD55
SPANLD56
SPANLD57
SPANLD58
SPANLD59
SPANLD60
SPANLD61
SPANLD62
SPANLD63
SPANLD64
SPANLD65
SPANLD66
SPANLD67
SPANLD68
SPANLD69
SPANLD70
SPANLD71
SPANLD72
SPANLD73
SPANLD74
SPANLD75
SPANLD76
SPANLD77
SPANLD78
SPANLD79
SPANLD80
SPANLD81
SPANLD82
SPANLD83
SPANLD84
SPANLD85
SPANLD86
SPANLD87
SPANLD88
SPANLD89
SPANLD90
SPANLD91
SPANLD92
SPANLD93
SPANLD94
SPANLD95
SPANLD96
SPANLD97
SPANLD98
SPANLD99
SPANL100
SPANL101

```

```

IF (NASYM.EQ.0) GO TO 200
CALL RDAERO (4HDPAP,IVBW,D,NDUT,NER)
DC 150 I=1,NBCX
DCP(I) = 0.0
DC 160 J=1,NASYM
JJ = J+NSYM
160 DCP(I) = DCP(I) + D(I,J)*Q(JJ)
C
IF (NSBE.EQ.0) GO TO 200
CALL RDAERO (4HDPAP,IVBW,DB,NDUT,NER)
DC 170 I=1,NSRE
FZ(I) = 0.0
DC 170 J=1,NASYM
JJ = J+NSYM
170 FZ(I) = FZ(I) + DB(I,J)*Q(JJ)
C
CALL RDAERO (4HDYAP,IVRW,DB,NDUT,NER)
DC 180 I=1,NSRE
FY(I) = 0.0
DC 180 J=1,NASYM
JJ = J+NSYM
180 FY(I) = FY(I) + DR(I,J)*Q(JJ)
C
CALL AERO2 (NDUT,INDSYM,DCP,FZ,FY
1 , A(L1),A(L2),A(L3),A(L4),A(L5),A(L6),A(L7),A(L8),A(L9),A(L10)
2 , A(L11),A(L12),A(L13),A(L14),A(L15),A(L16),A(L17),A(L18),A(L19)
3 , A(L20),A(L21),A(L22),A(L23),A(L24),A(L25),A(L26),A(L27),A(L28))
C 200 RETURN
END

```

SPANLI02
 SPANLI03
 SPANLI04
 SPANLI05
 SPANLI06
 SPANLI07
 SPANLI08
 SPANLI09
 SPANLI10
 SPANLI11
 SPANLI12
 SPANLI13
 SPANLI14
 SPANLI15
 SPANLI16
 SPANLI17
 SPANLI18
 SPANLI19
 SPANLI20
 SPANLI21
 SPANLI22
 SPANLI23
 SPANLI24
 SPANLI25
 SPANLI26
 SPANLI27
 SPANLI28
 SPANLI29
 SPANLI30
 SPANLI31
 SPANLI32
 SPANLI33
 SPANLI34

```

SUBROUTINE TRIM(AMODNO,EMBAR,CAY,NSYM,NASYM,NMODES,NINTLD,NMS,NMA
1  , Q,QDOT,ZSYM,FSYM,DSYM,ZASYM,FASYM,DASYM
2  , PS,PA,PIQ,PAQS,PAQA,THRLD,THRGNF,THRUST
3  , TVRW )
C
COMMON/ZZZ/CASE(48),NIN,NGUT,KROW,LINES,IPRNT,NFR
COMMON/XTF/NFLAGS(100)
C
EQUIVALENCE (NFLAGS(10),NENGS)
EQUIVALENCE (NFLAGS(12),BZERO), (NFLAGS(13),PH20)
1  , (NFLAGS(14),SIGMA), (NFLAGS(16),TUNITS), (NFLAGS(17),VKFAS)
1  , (NFLAGS(19),ALT)
2  , (NFLAGS(18),VTFPS)
2  , (NFLAGS(23),KPRCHK)
4  , (NFLAGS(41),NK), (NFLAGS(42),NG)
EQUIVALENCE (NFLAGS(51),AN), (NFLAGS(52),ZDOT), (NFLAGS(53),RTURN)
1  , (NFLAGS(54),KMAN), (NFLAGS(55),AH), (NFLAGS(56),AC)
1  , (NFLAGS(57),INDSYM)
3  , (NFLAGS(100),SIZECT)
4  , (NFLAGS(99),RBRADF)
C
DIMENSION ZSYM(NMS,NMS)
DIMENSION ZASYM(NMA,NMA)
DIMENSION VELUN(3)
DIMENSION EMBAR(NMODES,NMODES)
DIMENSION CAY(1)
DIMENSION FSYM(NMS)
DIMENSION FASYM(NMA)
DIMENSION Q(NMCDES)
DIMENSION QDOT(NMODES)
DIMENSION DSYM(2,NSYM,NSYM)
DIMENSION DASYM(2,NASYM,NASYM)
DIMENSION AMODNO(20)
DIMENSION PS(NINTLD)
DIMENSION PA(NINTLD)
DIMENSION PIQ(NINTLD,NMODES)
DIMENSION PAQS(2,NINTLD,NSYM)
DIMENSION PAQA(2,NINTLD,NASYM)
DIMENSION THRLD(NINTLD,1), THRGNF(NSYM,1)
DIMENSION THRUST(1)
C
COMMON/DISK2/ND2,ITBL2(843),NRECSA,IBUMP,NKD,VDBWS(20)
C
10 FORMAT(1H0,'*SINGULAR MATRIX FOR TVRW=*,I4,5X,*VORW=*,E16.8/1H0,
1 44  *,13(4H***))
11 FCPRMAT(1H0,30X,'*TRIMMED SYMMETRIC FLIGHT SOLNS*/1H0,
1 10X,R.A. PLUNGE MODE = ,I4/1H ,
1 10X,R.A. PITCH MODE = ,I4/1H ,
1 10X,SYMMETRIC TRIM MODE = ,I4/1H ,
1 10X,*SYM R.B. JIG MODE = ,I4/1H ,
1 10X,*1ST SYM ELASTIC MODE = ,I4/1H ,
1 10X,*LST SYM ELASTIC MODE = ,I4/1H ,

```



```

IF (VORWS(I).LT.VORWN) GO TO 98
VORWN = VORWS(I)
CONTINUE
98
N40 = 2*NSYM*NSYM
LCC=1+(IVBW-1)*IRUMP
CALL READMS(ND2,DSYM ,NWD,LCC)
FORM EQNS OF EQUILIBRIUM
DO 300 I=1,NMS
ESYM(I)=0.
DO 300 J=1,NMS
ZSYM(I,J)=0.
300
FORM Z MATRIX
MH=AMODNO(1)
MA=AMODNO(2)
MT=AMODNO(6)
MJ=AMODNO(7)
MEF=AMODNO(4)
MEL=AMODNO(5)
WRITE(ROUT,11) MH,MA,MT,MJ,MEF,MEL,VTFPS,DYNP,AN,QBARP
11
LINES=LINES+16
DELX=FMBAR(MH,MA)/FMBAR(MH,MH)
571
IF (KPRC-K.EQ.0) GO TO 571
CALL PRNT(DSYM,NSYM,NSYM,2,12HDSYMREAL AERO ,3)
CALL PRNT(EMBAR(1,MH),NSYM,1,NMODES,1,12HFMBAR(MH) ,3)
CONTINUE
IF (MT.FQ.0) GO TO 1015
ZSYM(1,1)=AMULTA*DSYM(1,MH,MA)
ZSYM(1,2)=AMULTA*DSYM(1,MH,MT)
ZSYM(2,1)=AMULTA*DSYM(1,MA,MA)
ZSYM(2,2)=AMULTA*DSYM(1,MA,MT)
IF (MEF.FQ.0) GO TO 311
IE=2
DO 310 I=MEF,MEL
IF=IE+1
ZSYM(1,IE)=AMULTA*DSYM(1,MH,I)
ZSYM(1,IE)=AMULTA*DSYM(1,I,MA)
ZSYM(2,IE)=AMULTA*DSYM(1,MA,I)
ZSYM(2,IE)=AMULTA*DSYM(1,I,MT)
ZSYM(IE,IE)=CAY(I)
310
IFF=2

```

TRIM 158
 TRIM 159
 TRIM 160
 TRIM 161
 TRIM 162
 TRIM 163
 TRIM 164
 TRIM 165
 TRIM 166
 TRIM 167
 TRIM 168
 TRIM 169
 TRIM 170
 TRIM 171
 TRIM 172
 TRIM 173
 TRIM 174
 TRIM 175
 TRIM 176
 TRIM 177
 TRIM 178
 TRIM 179
 TRIM 180
 TRIM 181
 TRIM 182
 TRIM 183
 TRIM 184
 TRIM 185
 TRIM 186
 TRIM 187
 TRIM 188
 TRIM 189
 TRIM 190
 TRIM 191
 TRIM 192
 TRIM 193
 TRIM 194
 TRIM 195
 TRIM 196
 TRIM 197
 TRIM 198
 TRIM 199
 TRIM 200
 TRIM 201
 TRIM 202
 TRIM 203
 TRIM 204
 TRIM 205
 TRIM 206
 TRIM 207
 TRIM 208
 TRIM 209

```

C 310 J=MEF,MEL
C 311 IFF=IEE+1
C 312 ZSYM(IE,IFF)=ZSYM(IE,IEE)+AMULTA*DSYM(1,I,J)
C 313 CONTINUE
C 314 FORM DO MATRIX
C 315 IF(MJ.EQ.0) GO TO 340
C 316 FSYM(1)=FSYM(1)-AMULTA*DSYM(1,MH,MJ)
C 317 FSYM(2)=FSYM(2)-AMULTA*DSYM(1,MA,MJ)
C 318 IE=2
C 319 IF(MEF.EQ.0) GO TO 331
C 320 DO 330 I=MEF,MEL
C 321 IFF=IE+1
C 322 FSYM(IE)=FSYM(IE)-AMULTA*DSYM(1,I,MJ)
C 323 CONTINUE
C 324 CONTINUE
C 325 IF(KMAN.EQ.0) GO TO 326
C 326 FOR NON ZERO QBAR, READ IN LOWEST K SYM AERO WITH NON 0 IMAG PARTS
C 327 LCC=1+(1VRWN-1)*IRUMP
C 328 CALL READMS(ND2,DSYM,NWD,LOC)
C 329 VCBWN=1/K FOR THIS AERC SET
C 330 IF(KPRCH-K.EQ.0) GO TO 572
C 331 CALL PRNT(DSYM,NSYM,NSYM,2,12HDSYMIMAGAERO,3)
C 572 CONTINUE
C 332 CALC VELOCITY AERC
C 333 AFAC=8ZER0*SIZECT*VQBWN/(12.*VTFPS)
C 334 FSYM(1)=-AMULTA*(DSYM(2,MH,MA)+DSYM(2,MH,MH)*DELX)*QBAR*AFAC
C 335 FSYM(1)+FSYM(1)
C 336 FSYM(2)=-AMULTA*(DSYM(2,MA,MA)+DSYM(2,MA,MH)*DELX)*QBAR*AFAC
C 337 FSYM(2)+FSYM(2)
C 338 IF(MEF.EQ.0) GO TO 326
C 339 IFF=2
C 340 DO 325 I=MEF,MEL
C 341 IFF=IE+1
C 342 FSYM(IE)=-AMULTA*(DSYM(2,I,MA)+DSYM(2,I,MH)*DELX)*QBAR*AFAC
C 343 FSYM(IE)+FSYM(IE)
C

```

```

TRIM 210
TRIM 211
TRIM 212
TRIM 213
TRIM 214
TRIM 215
TRIM 216
TRIM 217
TRIM 218
TRIM 219
TRIM 220
TRIM 221
TRIM 222
TRIM 223
TRIM 224
TRIM 225
TRIM 226
TRIM 227
TRIM 228
TRIM 229
TRIM 230
TRIM 231
TRIM 232
TRIM 233
TRIM 234
TRIM 235
TRIM 236
TRIM 237
TRIM 238
TRIM 239
TRIM 240
TRIM 241
TRIM 242
TRIM 243
TRIM 244
TRIM 245
TRIM 246
TRIM 247
TRIM 248
TRIM 249
TRIM 250
TRIM 251
TRIM 252
TRIM 253
TRIM 254
TRIM 255
TRIM 256
TRIM 257
TRIM 258
TRIM 259
TRIM 260
TRIM 261

```

```

326 CONTINUE
C
C
C
FSYM(1)=FSYM(1)+EMBAR(MH,MH)*ANGFE
FSYM(2)=FSYM(2)+EMBAR(MA,MH)*ANGEE
IF=2
IF(MEF.EQ.0)GO TO 351
DO 350 I=MEF,MEL
IF=IF+1
350 FSYM(IE)=FSYM(IE)+EMBAR(I,MH)*ANGEE
351 CONTINUE
C
C
C
ADD THRUST TERMS
IF(NFNGS.EQ.0)GO TO 450
READ IN THRUD AND THRGNF FROM UNIT DATASET
CALL ROUNIT (4,0,THPLD,NOUT,NER)
CALL ROUNIT (5,0,THRGNF,NOUT,NER)
DO 410 I=1,NFNGS
FSYM(1)=FSYM(1)+THRGNF(MH,I)*THRUST(I)
FSYM(2)=FSYM(2)+THRGNF(MA,I)*THRUST(I)
IF=2
IF(MEF.EQ.0)GO TO 409
DO 408 J=MEF,MEL
IF=IF+1
408 FSYM(IE) = FSYM(IE) + THRGNF(IE,I)*THRUST(I)
409 CONTINUE
410 CONTINUE
450 CONTINUE
C
C
IF(KPRCHK.EQ.0)GO TO 570
CALL PRNT(ZSYM,NMS,NMS,1,4HZSYM,1)
CALL PRNT(FSYM,NMS,1,NMS,1,4HFSYM,1)
WRITE(NOUT,60)IVBW
570 CONTINUE
C
DET=0.
CALL MSL(ZSYM,NMS,NMS,FSYM,1,NER,DET)
IF(NER.EQ.0)GO TO 680
WRITE(NOUT,10)IVBW,VORW
GO TO 5000
C
680 CONTINUE
PALP=FSYM(1)/(RBRADF*RAD)
PDLT=FSYM(2)/(RBRADF*RAD)
DO 690 I=1,NMDES
QDOT(I)=0.
Q(I)=0.
Q(MA)=FSYM(1)
QDOT(MA)=QBRAP
QDOT(MH)=DELX*QBAR
690

```

```

TRIM 262
TRIM 263
TRIM 264
TRIM 265
TRIM 266
TRIM 267
TRIM 268
TRIM 269
TRIM 270
TRIM 271
TRIM 272
TRIM 273
TRIM 274
TRIM 275
TRIM 276
TRIM 277
TRIM 278
TRIM 279
TRIM 280
TRIM 281
TRIM 282
TRIM 283
TRIM 284
TRIM 285
TRIM 286
TRIM 287
TRIM 288
TRIM 289
TRIM 290
TRIM 291
TRIM 292
TRIM 293
TRIM 294
TRIM 295
TRIM 296
TRIM 297
TRIM 298
TRIM 299
TRIM 300
TRIM 301
TRIM 302
TRIM 303
TRIM 304
TRIM 305
TRIM 306
TRIM 307
TRIM 308
TRIM 309
TRIM 310
TRIM 311
TRIM 312
TRIM 313

```

```

Q(MT)=FSYM(2)
IF(MJ.NE.0)Q(MJ)=1.
IF(MEF.EQ.0)GO TO 696
IF=2
DO 695 I=MEF,MEL
IE=IE+1
Q(I)=FSYM(IE)
C
695 CCONTINUE
C
LINES=LINES+4
IF(LINES.LT.KROW)GO TO 880
CALL HEADNG
WRITE(NOUT,50)VKEAS,VELUN(IUNIT),ALT
LINES=9
880 CONTINUE
WRITE(NOUT,40)
DO 890 JJ=1,NSYM
LINES=LINES+1
IF(LINES.LT.KROW)GO TO 895
CALL HEADNG
WRITE(NOUT,50)VKEAS,VELUN(IUNIT),ALT
LINES=9
895 CONTINUE
WRITE(NOUT,30)JJ,Q(JJ),QDOT(JJ)
900 CONTINUE
C
WRITE(NOUT,4)PALPH,POELT
LINES=LINES+2
C
C
C
IF(INDSYM.EQ.1)GO TO 2001
READ IN LOWST K GENLZD ASYM AERO INTO DASYM
C
C
C
NWD = 2*NASYM*NASYM
LOC=2+(I*VBW-1)*I*BJUMP
CALL READMS(ND2,DASYM,NWD,LOC)
C
C
C
VCBW=1/K FOR THIS AERO SET
C
FORM EQNS OF EQUILIBRIUM
C
DO 1300 I=1,NMA
EASYM(I)=0.
DO 1300 J=1,NMA
ZASYM(I,J)=0.
1300 C
C
C
FORM Z MATPIX
MR=AMQDNC(11)
C
C
C

```

TRIM 366
 TRIM 367
 TRIM 368
 TRIM 369
 TRIM 370
 TRIM 371
 TRIM 372
 TRIM 373
 TRIM 374
 TRIM 375
 TRIM 376
 TRIM 377
 TRIM 378
 TRIM 379
 TRIM 380
 TRIM 381
 TRIM 382
 TRIM 383
 TRIM 384
 TRIM 385
 TRIM 386
 TRIM 387
 TRIM 388
 TRIM 389
 TRIM 390
 TRIM 391
 TRIM 392
 TRIM 393
 TRIM 394
 TRIM 395
 TRIM 396
 TRIM 397
 TRIM 398
 TRIM 399
 TRIM 400
 TRIM 401
 TRIM 402
 TRIM 403
 TRIM 404
 TRIM 405
 TRIM 406
 TRIM 407
 TRIM 408
 TRIM 409
 TRIM 410
 TRIM 411
 TRIM 412
 TRIM 413
 TRIM 414
 TRIM 415
 TRIM 416
 TRIM 417

```

MY=AMODNO(12)
ML=AMODNO(13)
MJA=AMODNO(18)
MEFA=AMODNO(14)
MELA=AMODNO(15)
MTR=AMODNO(16)
MTY=AMODNO(17)
LINES=LINES+18
IF(LINES.GE.KROW)CALL HEADNG
WRITE(NOUT,111)MR,ML,MY,MTR,MTY,MJA,MEFA,MELA,VTFPS,DYNP,AB,AC,
1 PRARP,R3ARP,RTURN
DELZ=EMBAR(MR,ML)/EMBAR(ML,ML)

IF(KPRCH-K.EQ.0)GO TO 573
CALL PRNT(DASYM,NASYM,NASYM,2,12HASYMREALAERO ,3)
CONTINUE
573

IF(MTR.FQ.0)GO TO 1015
IF(MTY.FQ.0)GO TO 1015

MR = MR - NSYM
MY = MY - NSYM
ML = ML - NSYM
MJA=MJA- NSYM
MEFA=MEFA- NSYM
MELA=MELA- NSYM
MTR=MTR - NSYM
MTY=MTY - NSYM

ZASYM(1,1)=AMULTA*DASYM(1,ML,MY)
ZASYM(1,2)=AMULTA*DASYM(1,ML,MTY)
ZASYM(1,3)=AMULTA*DASYM(1,ML,MTR)
ZASYM(2,1)=AMULTA*DASYM(1,MR,MY)
ZASYM(2,2)=AMULTA*DASYM(1,MR,MTY)
ZASYM(3,1)=AMULTA*DASYM(1,MY,MY)
ZASYM(3,2)=AMULTA*DASYM(1,MY,MTY)
ZASYM(3,3)=AMULTA*DASYM(1,MY,MTR)
ZASYM(2,3)=AMULTA*DASYM(1,MR,MTR)

IF(MEFA.LE.0)GO TC 1311
IF=3
DC 1310 I=MEFA,MELA
IF=IE+1
ZASYM(1,IF)=AMULTA*DASYM(1,ML,I)
ZASYM(1,IF)=AMULTA*DASYM(1,I,MY)
ZASYM(2,IF)=AMULTA*DASYM(1,MR,I)
ZASYM(2,IF)=AMULTA*DASYM(1,I,MTY)
ZASYM(3,IF)=AMULTA*DASYM(1,MY,I)
ZASYM(3,IF)=AMULTA*DASYM(1,I,MTR)
IF=I+NSYM
ZASYM(IF,IE)=CAY(II)
  
```

```

C      IF I=3
C      DO 1310 J=MEFA,MFLA
C      IF I=IE+1
1310 7ASYM(IF,IEE)=ZASYM(IE,IEE)+AMULTA*DASYM(1,I,J)
C      1311 CONTINUE
C      FORM CO MATRIX
C      IF (MJA.LE.0) GO TO 1340
C      FASYM(1)=FASYM(1)-AMULTA*DASYM(1,ML,MJA)
C      FASYM(2)=FASYM(2)-AMULTA*DASYM(1,MR,MJA)
C      FASYM(3)=FASYM(3)-AMULTA*DASYM(1,MY,MJA)
C      IE=3
C      IF (MEFA.EQ.0) GO TO 1331
C      DO 1330 I=MEFA,MELA
C      IF I=IE+1
1330 FASYM(IF)=FASYM(IE)-AMULTA*DASYM(1,I,MJA)
C      1331 CONTINUE
C      1340 CONTINUE
C      READ IN LOWEST K ASYM AERO WITH NON 0 IMAG PARTS
C      VOBWN=1/K FOR THIS AERO SET
C      LCC=2+(1/VBWN-1)*IBUMP
C      CALL READMS(ND2,DASYM,NWD,LOC)
C      IF (KPRCH-K.EQ.0) GO TO 574
C      CALL PRNT(DASYM,NASYM,NASYM,2,12,HASYM,IMAGAERO ,3)
C      574 CONTINUE
C      CALC VELOCITY AERO
C      AFAC TA=B ZERO*SIZECT*VOBWN/(12.*VTFPS)
C      FASYM(1)=-AMULTA*(DASYM(2,ML,MR)+DASYM(2,ML,ML)*DELZ)*PBAR*AFAC TA
C      1 +FASYM(1)
C      FASYM(2)=-AMULTA*(DASYM(2,MR,MR)+DASYM(2,MR,ML)*DELZ)*PBAR*AFAC TA
C      1 -AMULTA*(DASYM(2,MR,MY)-DASYM(2,MR,ML)*DELX)*PBAR*AFAC TA
C      1 +FASYM(2)
C      FASYM(3)=-AMULTA*(DASYM(2,MY,MR)+DASYM(2,MY,ML)*DELZ)*PBAR*AFAC TA
C      1 -AMULTA*(DASYM(2,MY,MY)-DASYM(2,MY,ML)*DELX)*PBAR*AFAC TA
C      1 +FASYM(3)

```

```

TRIM 418
TRIM 419
TRIM 420
TRIM 421
TRIM 422
TRIM 423
TRIM 424
TRIM 425
TRIM 426
TRIM 427
TRIM 428
TRIM 429
TRIM 430
TRIM 431
TRIM 432
TRIM 433
TRIM 434
TRIM 435
TRIM 436
TRIM 437
TRIM 438
TRIM 439
TRIM 440
TRIM 441
TRIM 442
TRIM 443
TRIM 444
TRIM 445
TRIM 446
TRIM 447
TRIM 448
TRIM 449
TRIM 450
TRIM 451
TRIM 452
TRIM 453
TRIM 454
TRIM 455
TRIM 456
TRIM 457
TRIM 458
TRIM 459
TRIM 460
TRIM 461
TRIM 462
TRIM 463
TRIM 464
TRIM 465
TRIM 466
TRIM 467
TRIM 468
TRIM 469

```

```

C
IF (MEFA, LF, 0) GO TO 1326
IF=3
OC 1325 I=MEFA, MEFA
IF=IF+1
1325 FASYM(IE)=-AMULTA*(DASYM(2, I, MR)+DASYM(2, I, ML)*DELZ)*PBAR*AFAC TA
1 -AMULTA*(DASYM(2, I, MY)-DASYM(2, I, ML)*DELX)*RBAR*AFAC TA
1 +FASYM(IE)
C
1326 CONTINUE
C
C
C
C
C
1570 CONTINUE
IF (KPRCHK, EQ, 0) GO TO 1570
CALL PRNT(ZASYM, NMA, NMA, NMA, 1, 8H ZASYM , 2)
CALL PRNT(FASYM, NMA, 1, NMA, 1, 8H FASYM , 2)
WRITE (NCUT, 60) IVBW
1570 CONTINUE
C
DET=0
CALL MTS1(ZASYM, NMA, NMA, FASYM, 1, NER, DET)
IF (NER, EQ, 0) GO TO 1680
WRITE (NCUT, 10) IVBW, VORW
C
1680 CONTINUE
PRETA=FASYM(1)/(RBRADF*RAD)
PDELY=FASYM(2)/(RBRADF*RAD)
PDELPR=FASYM(3)/(RBRADF*RAD)
C
MP = MR + NSYM
MY = MY + NSYM
ML = ML + NSYM
MJA = MJA + NSYM
MFA=MEFA+ NSYM
MELA=MELA+ NSYM
MTY=MTY + NSYM
MTR=MTR + NSYM
Q(MY)=FASYM(1)
Q(MTY)=FASYM(2)
Q(MTR)=FASYM(3)
COUT(MR)=PRAP
COUT(ML)=DELZ*PBAR -DELX*RBAR
COUT(MY)=RBAR
IF (MJA, NE, 0) Q(MJA)=1.
IF (MEFA, EQ, 0) GO TO 1696
IF=3
OC 1695 I=MEFA, MEFA
IF=IF+1
1695 Q(1)=FASYM(IE)
C

```

TRIM 470
 TRIM 471
 TRIM 472
 TRIM 473
 TRIM 474
 TRIM 475
 TRIM 476
 TRIM 477
 TRIM 478
 TRIM 479
 TRIM 480
 TRIM 481
 TRIM 482
 TRIM 483
 TRIM 484
 TRIM 485
 TRIM 486
 TRIM 487
 TRIM 488
 TRIM 489
 TRIM 490
 TRIM 491
 TRIM 492
 TRIM 493
 TRIM 494
 TRIM 495
 TRIM 496
 TRIM 497
 TRIM 498
 TRIM 499
 TRIM 500
 TRIM 501
 TRIM 502
 TRIM 503
 TRIM 504
 TRIM 505
 TRIM 506
 TRIM 507
 TRIM 508
 TRIM 509
 TRIM 510
 TRIM 511
 TRIM 512
 TRIM 513
 TRIM 514
 TRIM 515
 TRIM 516
 TRIM 517
 TRIM 518
 TRIM 519
 TRIM 520
 TRIM 521

TRIM 522
TRIM 523
TRIM 524
TRIM 525
TRIM 526
TRIM 527
TRIM 528
TRIM 529
TRIM 530
TRIM 531
TRIM 532
TRIM 533
TRIM 534
TRIM 535
TRIM 536
TRIM 537
TRIM 538
TRIM 539
TRIM 540
TRIM 541
TRIM 542
TRIM 543
TRIM 544
TRIM 545
TRIM 546
TRIM 547
TRIM 548
TRIM 549
TRIM 550
TRIM 551
TRIM 552
TRIM 553
TRIM 554
TRIM 555
TRIM 556
TRIM 557
TRIM 558
TRIM 559
TRIM 560
TRIM 561
TRIM 562
TRIM 563
TRIM 564
TRIM 565
TRIM 566
TRIM 567
TRIM 568
TRIM 569
TRIM 570
TRIM 571
TRIM 572
TRIM 573

```

1696 C CONTINUE
      LINES=LINES+2
      IF (LINES.LT.KROW) GO TO 1880
      CALL HEADNG
      WRITE(NOUT,50) VKFAS,VELUN(IUNIT),ALT
      LINES=9
1880 C CONTINUE
      WRITE(NOUT,40)
      DO 1890 J=1,NASYN
      JJ=J+NSYN
      LINES=LINES+1
      IF (LINES.LT.KROW) GO TO 1895
      CALL HEADNG
      WRITE(NOUT,50) VKFAS,VELUN(IUNIT),ALT
      LINES=9
1895 C CONTINUE
      WRITE(NOUT,30) JJ,Q(JJ),QDOT(JJ)
1890 C CONTINUE
      LINES=LINES+3
2001 C CONTINUE
      READ IN PIQ
      CALL ROUNIT (S,O,PIQ,NOUT,NER)
      IF (NER.NE.O) GO TO 5000
      IF (KPRCHK.EQ.O) GO TO 108
      CALL PRNT (PIQ,NINTLD,NMODES,NINTLD,1,4HP IQ ,1)
108 C CONTINUE
      DO 110 I=1,NINTLD
      PS(I)=0.
      PA(I)=0.
      PS(I)=PS(I)+PIQ(I,MH)*ANGLEE
      IF (NENG.S.EQ.O) GO TO 110
      DO 109 J=1,NENG
      PS(I)=PS(I)+THPLD(I,J)*THRUST(J)
109 C CONTINUE
110 C
      CALC DYNP X MOTION DEPENDENT AERO FORCES X Q
C
C
C
C
      READ IN PAQS(L) AND PAQA(L) FOR K=0 (VCBWSM AND VCBWAM)
      CALL ROUNIT (7,IVBW,PAQS,NOUT,NER)
      IF (INDSYM.EQ.1) GO TO 112
      CALL ROUNIT (8,IVRW,PAQA,NOUT,NER)

```

TRIM	574
TRIM	575
TRIM	576
TRIM	577
TRIM	578
TRIM	579
TRIM	580
TRIM	581
TRIM	582
TRIM	583
TRIM	584
TRIM	585
TRIM	586
TRIM	587
TRIM	588
TRIM	589
TRIM	590
TRIM	591
TRIM	592
TRIM	593
TRIM	594
TRIM	595
TRIM	596
TRIM	597
TRIM	598
TRIM	599
TRIM	600
TRIM	601
TRIM	602
TRIM	603
TRIM	604
TRIM	605
TRIM	606
TRIM	607
TRIM	608
TRIM	609
TRIM	610
TRIM	611
TRIM	612
TRIM	613
TRIM	614
TRIM	615
TRIM	616
TRIM	617
TRIM	618
TRIM	619
TRIM	620
TRIM	621
TRIM	622
TRIM	623
TRIM	624
TRIM	625

TRIM 626
 TRIM 627
 TRIM 628
 TRIM 629
 TRIM 630
 TRIM 631
 TRIM 632
 TRIM 633
 TRIM 634
 TRIM 635
 TRIM 636
 TRIM 637
 TRIM 638
 TRIM 639
 TRIM 640
 TRIM 641
 TRIM 642
 TRIM 643
 TRIM 644
 TRIM 645
 TRIM 646
 TRIM 647
 TRIM 648
 TRIM 649
 TRIM 650
 TRIM 651
 TRIM 652
 TRIM 653

```

    WRITE (NCUT,2)
  2  FORMAT (1H0,20X,*SYM INTEGRATED LOADS FOR TRIM*)
    CALL PRNT(PS,NINTLD,1,NINTLD,1,4H
    ,1)
    IF (INDSYM.EQ.1) GO TO 280
    WRITE (NCUT,3)
  3  FORMAT (1H0,20X,*ASYM INTEGRATED LOADS FOR TRIM*)
    CALL PRNT(PA,NINTLD,1,NINTLD,1,4H
    ,1)
  280 CONTINUE
C
C      NOW SAVE PS AND PA FOR SURSEQUENT USE
C
C      5000 RETURN
C
  1015 CONTINUE
    WRITE (NCUT,1016)
  1016 FORMAT(1H1,20X,*CANNOT TRIM,NO TRIM MODE SPECIFIED*)
    STOP
C
  1017 CONTINUE
    WRITE (NCUT,1018)AN
  1018 FORMAT(1H1,20X,HEY TRY SOME LOAD FACTOR OTHER THAN ,F8.4,
    1  FOR THE TURN )
    STOP
C
    END
  
```

```

SUBROUTINE ACCHST
1 C FTG,NACC,NREQ,NTRSP,OMEGMX,INCSYM,OMEGA
2 C ,AS,AC,AST,AAT,TIME,TIMEMX,KPHLS,NG,NGL,IPLBL,PLOT)
COMMON /Z77/CASE(48),NIN,NOUT,KROW,LINFS,IPRNT,NFP
DIMENSION FTG(2,1), TIME(1), PLOT(1)
DIMENSION AS(2,NACC,NREQ), AA(2,NACC,NREQ)
DIMENSION AST(NACC,1), AAT(NACC,1)
DATA TWOPI/.2831853/
1) FORMAT (1H0,20X,*SYMMETRIC TIME RESPONSE - INCREMENTAL ACCELERATION*
11) FORMAT (1H0,20X,*UNSYMMETRIC TIME RESPONSE - INCREMENTAL ACCELERATION*
11) FORMAT (1H0,20X,FR.4,*RESPONSE SECS*,2X,FR.4,*HZ COUNTED*)
40) FORMAT (1H0,2X,*TIME **,8(7X,*STA*,13))
50) FORMAT (1H,FR.4,8(1X,E12.5))
INDSYM=1 IS A SYMMETRIC CASE, INDSYM=2, UNSYMMETRIC
DO 100 J=1,NREQ
AR = FTG(1,J)
AT = FTG(2,J)
DO 100 I=1,NACC
T = AS(1,I,J)*AR - AS(2,I,J)*AT
AS(2,I,J) = AS(1,I,J)*AT + AS(2,I,J)*AR
AC(1,I,J) = T
IF (INDSYM.EQ.1) GO TO 100
T = AA(1,I,J)*AR - AA(2,I,J)*AT
AA(2,I,J) = AA(1,I,J)*AT + AA(2,I,J)*AR
AA(1,I,J) = T
100 CONTINUE
CALL IF1 (NREQ,OMEGA,OMEGMX,0.,0,NACC,AS,NTRSP,TIME,AST,TIMEMX)
IF (INDSYM.EQ.1) GO TO 150
CALL IF2 (NREQ,OMEGA,OMEGMX,0.,0,NACC,AA,NTRSP,TIME,AAT,TIMEMX)
150 CONTINUE
AMA = 1.0
IF (NGL.LT.0) AMA = -1.0
DO 500 I=1,NACC
IF (INDSYM.EQ.1) GO TO 500
CALC LHS AND RHS ACC. (UNSYMMETRIC LOAD CONDITION)

```

ACCHST 2
ACCHST 3
ACCHST 4
ACCHST 5
ACCHST 6
ACCHST 7
ACCHST 8
ACCHST 9
ACCHST10
ACCHST11
ACCHST12
ACCHST13
ACCHST14
ACCHST15
ACCHST16
ACCHST17
ACCHST18
ACCHST19
ACCHST20
ACCHST21
ACCHST22
ACCHST23
ACCHST24
ACCHST25
ACCHST26
ACCHST27
ACCHST28
ACCHST29
ACCHST30
ACCHST31
ACCHST32
ACCHST33
ACCHST34
ACCHST35
ACCHST36
ACCHST37
ACCHST38
ACCHST39
ACCHST40
ACCHST41
ACCHST42
ACCHST43
ACCHST44
ACCHST45
ACCHST46
ACCHST47
ACCHST48
ACCHST49
ACCHST50
ACCHST51
ACCHST52
ACCHST53

```

C 200 J=1,NTMRSP
TEMP = AST(I,J) - AMA*AAAT(I,J)
AST(I,J) = AST(I,J) + AMA*AAAT(I,J)
200 AAAT(I,J) = TEMP
C 500 CONTINUE
C
C IF (IPLBL.EQ.0) GO TO 800
NEW PLOT ACCELERATION TIME HISTORIES
C
C 750 I=1,NACC
IF (IPLBL.EQ.2) GO TO 720
C
C 710 J=1,NTMRSP
PLOT(J) = AST(I,J)
CALL PLT1 (NTMRSP, TIME, PLCT, 1)
WRITE (NCUT,21) I,NG
21 FORMAT (1H0,20X,*ACCELERATION TIME HISTORY RHS ACC. NO.*,I4,4X
1,*AND ORIENTATION NO.*,I4)
C 720 IF (IPLBL.EQ.1) GO TO 750
IF (INDSYM.EQ.1) GO TO 750
C
C 730 J=1,NTMRSP
PLOT(J) = AAT(I,J)
CALL PLT1 (NTMRSP, TIME, PLCT, 1)
WRITE (NCUT,22) I,NG
22 FORMAT (1H0,20X,*ACCELERATION TIME HISTORY LHS ACC. NO.*,I4,4X
1,*AND ORIENTATION NO.*,I4)
C 750 CONTINUE
C
C 800 IF (KPRBL.EQ.0) GO TO 1000
FRQMX = OMEGMAX/TWOPI
LINES = KRCW+1
C
C 820 JS=1,NACC,8
JF = MIN0(JS+7,NACC)
IF (LINES+3.LE.KRCW) GO TO 810
CALL HEADNG
WRITE (NCUT,10) TIMEFX,FRQMX
LINES = LINES+4
WRITE (NCUT,40) (J,J=JS,JF)
LINES = LINES+2
C 820 K=1,NTMRSP
LINES = LINES+1
IF (LINES.LE.KRCW) GO TO 820
CALL HEADNG
WRITE (NCUT,40) (J,J=JS,JF)

```

ACCHSI54
 ACCHSI55
 ACCHSI56
 ACCHSI57
 ACCHSI58
 ACCHSI59
 ACCHSI60
 ACCHSI61
 ACCHSI62
 ACCHSI63
 ACCHSI64
 ACCHSI65
 ACCHSI66
 ACCHSI67
 ACCHSI68
 ACCHSI69
 ACCHSI70
 ACCHSI71
 ACCHSI72
 ACCHSI73
 ACCHSI74
 ACCHSI75
 ACCHSI76
 ACCHSI77
 ACCHSI78
 ACCHSI79
 ACCHSI80
 ACCHSI81
 ACCHSI82
 ACCHSI83
 ACCHSI84
 ACCHSI85
 ACCHSI86
 ACCHSI87
 ACCHSI88
 ACCHSI89
 ACCHSI90
 ACCHSI91
 ACCHSI92
 ACCHSI93
 ACCHSI94
 ACCHSI95
 ACCHSI96
 ACCHSI97
 ACCHSI98
 ACCHSI99
 ACCHSI00
 ACCHSI01
 ACCHSI02
 ACCHSI03
 ACCHSI04
 ACCHSI05

ACCHSI06
ACCHSI07
ACCHSI08
ACCHSI09
ACCHSI10
ACCHSI11
ACCHSI12
ACCHSI13
ACCHSI14
ACCHSI15
ACCHSI16
ACCHSI17
ACCHSI18
ACCHSI19
ACCHSI20
ACCHSI21
ACCHSI22
ACCHSI23
ACCHSI24
ACCHSI25
ACCHSI26
ACCHSI27
ACCHSI28
ACCHSI29

```

      LINES = LINES+3
      WRITE (NOUT,50) TIME(K), (AST(J,K),J=JS,JF)
      IF (INDSYM.EQ.1) GO TO 1000
      DO 920 JS=1,NACC,R
      JF = MINO(JS+7,NACC)
      IF (LINES+3.LE.KROW) GO TO 910
      CALL HEADING
      WRITE (NOUT,11)
      WRITE (NOUT,12) TIME(K),FROMX
      LINES = LINES+4
      910 WRITE (NOUT,40) (J,J=JS,JF)
      LINES = LINES+2
      DO 920 K=1,NTRSP
      LINES = LINES+1
      IF (LINES.LE.KROW) GO TO 920
      CALL HEADING
      WRITE (NOUT,40) (J,J=JS,JF)
      LINES = LINES+3
      920 WRITE (NOUT,50) TIME(K), (AAT(J,K),J=JS,JF)
      1000 RETURN
      END

```

```

SUBROUTINE CGUST (CO,AMODNO,EMBAR,CAY)
DIMENSION AMODNO(1),EMBAR(1),CAY(1),CR(1)
COMMON NAA, A(1)
COMMON /777/HEDR(48),NIN,NOUT,KROW,LINES,IPRNT,NER
COMMON /XTE/NF(100)
EQUIVALENCE (NF(5),NFREQ), (NF(32),NINTLD)
EQUIVALENCE (NF(51),AN), (NF(52),ZDXT), (NF(53),RTURN)
EQUIVALENCE (NF(61),NCRMAX), (NF(62),TIMEWX), (NF(63),EFR)
1, (NF(33),NSTRSS)
1, (NF(10),NENGS)
1, (NF(66),HGRD)
1, (NF(43),NBOX), (NF(44),NSBETO)
1, (NF(97),DELT)
1, (NF(29),NACC)

DATA CORERQ/8H PLAST /

L1 = NAA
L2 = L1 + NINTLD
L21 = L2 + NINTLD
L22 = L21 + NSTRSS*NINTLD
L23 = L22 + NENGS
L24 = L23 + NINTLD
NAA = L24 + NINTLD

READ STRESS MATRIX FROM UNIT DATASET IF NEEDED
IF(NSTRSS.NE.0)CALL RDUNIT(3,0,A(L21),NOUT,NER)

READ IN SYMCOO FROM UNIT LOAD TAPE
CALL RDUNIT(9,0,A(L23),NOUT,NER)
READ IN ASMCOO FROM UNIT LOAD TAPF
CALL RDUNIT(10,0,A(L24),NOUT,NER)

NEW READ IN ENGINE THRUSTS IF ANY DEFINED
IF(NENGS.EQ.0)GO TO 100
I1=L22
I2=L22+NENGS-1
READ IN THRUST
READ(NIN,310) (A(I),I=I1,I2)
100 CONTINUE

CALL CTRIM(AMODNO,EMBAR,CAY,A(L1),A(L2),A(L22))

L3 = NAA
L4 = L3 + NCRMAX
L5 = L4 + NCRMAX
L6 = L5 + NINTLD*8

```

CGUST 2
CGUST 3
CGUST 4
CGUST 5
CGUST 6
CGUST 7
CGUST 8
CGUST 9
CGUST 10
CGUST 11
CGUST 12
CGUST 13
CGUST 14
CGUST 15
CGUST 16
CGUST 17
CGUST 18
CGUST 19
CGUST 20
CGUST 21
CGUST 22
CGUST 23
CGUST 24
CGUST 25
CGUST 26
CGUST 27
CGUST 28
CGUST 29
CGUST 30
CGUST 31
CGUST 32
CGUST 33
CGUST 34
CGUST 35
CGUST 36
CGUST 37
CGUST 38
CGUST 39
CGUST 40
CGUST 41
CGUST 42
CGUST 43
CGUST 44
CGUST 45
CGUST 46
CGUST 47
CGUST 48
CGUST 49
CGUST 50
CGUST 51
CGUST 52
CGUST 53


```
L9 = MAXO(L9,L8+2*NACC*NREQ)
L10 = L9 + NTMGST
L11 = L10 + MAXO(NTMGST,NTMRSP)
L12 = L11 + NTMGST
L13 = L12 + NREQ
L14 = L13 + NREQ
L15 = L14 + 2*NREQ
L16 = L15 + NINTLD*NTMRSP
L17 = MAXO(L16,L15+NACC*NTMRSP)
L18 = L17 + NTMPSP
L19 = L18 + MAXO(NINTLD*8,NSTRSS*8)
LMAX = L19 + 2*NINTLD

WRITE(NOUT,999)LMAX,CORERQ
999 FORMAT(IH0,20X,I10,' WORDS OF CORE REQ FOR STEP ++ ,A10, +++ )

CALL GUSTOR(CR,NTMGST
1 , A(L1),A(L2),A(L3),A(L4)
2 , A(L9),A(L10),A(L11),A(L6),A(L12),A(L13)
3 , A(L7),A(L8),A(L15),A(L16)
4 , A(L1),A(L7),A(L8),A(L15),A(L16),A(L17),A(L18),A(L5),NTMRSP,
5 A(L7),A(L8))

300 FORMAT (A1I2)
310 FORMAT (6F12.0)

C
C
C
C
C
```

[illegible]

FLTP0554
FLTP0555
FLTP0556
FLTP0557
FLTP0558
FLTP0559
FLTP0560
FLTP0561
FLTP0562
FLTP0563
FLTP0564
FLTP0565
FLTP0566
FLTP0567
FLTP0568
FLTP0569
FLTP0570
FLTP0571
FLTP0572
FLTP0573
FLTP0574
FLTP0575
FLTP0576
FLTP0577
FLTP0578
FLTP0579
FLTP0580
FLTP0581
FLTP0582
FLTP0583
FLTP0584
FLTP0585
FLTP0586
FLTP0587
FLTP0588
FLTP0589
FLTP0590
FLTP0591
FLTP0592
FLTP0593
FLTP0594
FLTP0595
FLTP0596
FLTP0597
FLTP0598
FLTP0599
FLTP0600
FLTP0601
FLTP0602
FLTP0603
FLTP0604
FLTP0605

```

C      IF (LRURST.EQ.0) GO TO 1000
C      XP=SLNTRC*GAMX
C      YB=SLNTRC*GAMY
C      ZR=SLNTRC*GAMZ+ALT
C      GO TO 1000
C      300 CONTINUE
C      LEVEL FLIGHT
C      XE=-VFL*Y
C      YE=0.
C      ZE=ALT
C      IF (LRURST.EQ.0) GO TO 1000
C      XR=SLNTRC*GAMX
C      YB=SLNTRC*GAMY
C      ZR=SLNTRC*GAMZ+ALT
C      GO TO 1000
C      1000 CONTINUE
C      SLNTRG=SQRT((XE-XR)**2+(YE-YB)**2+(ZE-ZR)**2)
C      SLNTRC=-SLNTRG
C      DX=XE-XR
C      DY=YB-YB
C      DZ=ZE-ZR
C      IF (LRURST.NE.0)
C      1 WRITE (NOUT,1) T,XE,YE,ZE,XR,YB,ZR,
C      1 SLNTRD,SLNTRG,DX,DY,DZ
C      1 FORMAT(1H0,20X, COORDINATES OF AIRCRAFT AND BURST AT TIME = ,
C      1 E12.4, SECS /1H0,
C      1 1H0,20X, AIRCRAFT (EFAS) /
C      1 1H,20X, X = ,E12.5/
C      1 1H,20X, Y = ,E12.5/
C      1 1H,20X, Z = ,E12.5/
C      1 1H0,20X, BURST (EFAS) /
C      1 1H,20X, X = ,E12.5/
C      1 1H,20X, Y = ,E12.5/
C      1 1H,20X, Z = ,E12.5/
C      1 1H0,20X, DISTANCE BURST TO AIRCRAFT AT INTERCEPT /
C      1 1H,20X, SLNTRD = ,E12.5,
C      1 1H0,20X, DISTANCE BURST TO AIRCRAFT NOW IS /
C      1 1H,20X, SLNTRG = ,E12.5,
C      1 1H,20X, SLNTRG = ,E12.5,

```

FLTP0106
 FLTP0107
 FLTP0108
 FLTP0109
 FLTP0110
 FLTP0111
 FLTP0112
 FLTP0113
 FLTP0114
 FLTP0115
 FLTP0116
 FLTP0117
 FLTP0118
 FLTP0119
 FLTP0120
 FLTP0121

```

1 14 ,20X, XE-XB = ,E12.5, FT. /
1 14 ,20X, YE-YB = ,F12.5, FT. /
1 14 ,20X, ZE-ZB = ,E12.5, FT. )

C IF (URPOST.EQ.0) GO TO 2000
C IF (ZR.GE.HGRD) GO TO 2000

ZRG=HGRD-78
WETFE(NGUT,2)/BBG
FCPMAT(110,20X, *** ATTENTION *** BURST IS ,F10.2,
2 1 FEET BELOW GROUND LEVEL )

C 2000 CONTINUE
C RETURN
END

```


GSTH ST54
GSTH ST55
GSTH ST56
GSTH ST57
GSTH ST58
GSTH ST59
GSTH ST60
GSTH ST61
GSTH ST62
GSTH ST63
GSTH ST64
GSTH ST65
GSTH ST66
GSTH ST67
GSTH ST68
GSTH ST69
GSTH ST70
GSTH ST71
GSTH ST72
GSTH ST73
GSTH ST74
GSTH ST75
GSTH ST76
GSTH ST77
GSTH ST78
GSTH ST79
GSTH ST80
GSTH ST81
GSTH ST82
GSTH ST83
GSTH ST84
GSTH ST85
GSTH ST86
GSTH ST87
GSTH ST88
GSTH ST89
GSTH ST90
GSTH ST91
GSTH ST92
GSTH ST93
GSTH ST94
GSTH ST95
GSTH ST96
GSTH ST97
GSTH ST98
GSTH ST99
GSTH ST00
GSTH ST01
GSTH ST02
GSTH ST03
GSTH ST04
GSTH ST05

```

30  CONTINUE
    IF (CHR.LT.2.4) GO TO 35
    SCGTP=1.569*SHB**2.5
    GO TO 40

35  CONTINUE
    CALL INTLDB(SHB,0.,0.,14,0,0,SHRT,0.,0.,CSGCTP,SGGTP)
    CONTINUE
    SH=AMINI(CHR,1.1)
    CALL INTPPB(SH,0.,0.,16,0,0,SHRTA,0.,0.,CGTP,CTP)
    SCGTP=SGCT(SCCTP**2+SHB**2)

50  CONTINUE
    RTA=SCALP*SLNTRD
    RT=1.
    IF (RTA.GT.0.55) PR=1.0+0.06*(RTA-0.55)**(1./3.)
    RT=RTA**PR

    CALL PRESS(RP,7ETA)
    DELPR=ZETA
    DELP=ZETA*PO
    TARR=0.
    CALL TAP(RTA,TARR)
    TARR=-TARR*SCALT
    IF (KPRTMH.EQ.0) GO TO 60
    WRITE (NOUT,5) 7ETA,AMACH,RTA,PO,DELP,SCALR,SCALT
    FORMAT(1H,7ETA=,E12.5/1H,
1  MACH=,F8.4/1H,
1  RTA=,E16.5/1H,
1  PO=,F12.5, PSI /1H,
1  DELP=,F12.5, PSI /1H,
1  SCALR=,E12.5/1H,
1  SCALT=,E12.5)
    FORMAT(1H,20X,
4  1  TIME BACK TO BURST = ,F10.4, SECS )

60  CONTINUE
    WRITE (NOUT,4) TARR
    TAO=0
    TCONT=0
    T=0.0
    IF (KPRTMH.EQ.0) GO TO 90
    CALL HEADING
    WRITE (NOUT,2)
    FORMAT(1H,10X, GUST TIME HISTORY AT AAS, ORIGIN /1H0,
1  1  IT,5X, TIME ,2X, VELOCITY ,6X, DENSITY ,7X, SLNTRG )

```

```

C 99 CONTINUE
C
C 100 I=1,NTMGST
C ICNT=ICNT+1
C
C TIMG(I) = T-TAU
C
C CALL FLTPDS(T,SLNTRG,VEL,ALT,AN,AR,AC,GAMX,GAMY,KMAN,
C 1 SLNTRG,XR,YR,ZR,XE,YE,ZE)
C
C IF(KGRDT.EQ.0)GO TO 80
C SHA=SCALP*(7E-HGRD)
C SCA=SCALP*SQRT((XE-XR)**2+(YE-YR)**2)
C CONTINUE
C 80 TIMTOT=T-TARP
C
C CALL TPEVAL(TIMTOT,RHCA,VSS,SLNTRG,SHA,SHR,SGA,SGOP,KGRDT,
C 1 SRGOTP,CTP,SCALR,SCALT,GUSTV,RHOR)
C
C IF(IND.NE.0)GO TO 95
C IF(GUSTV.GT.0)GO TO 95
C IF(GUSTV.LE.VELGL)GO TO 95
C DELTG=10.*DELTG
C IAP=1
C CONTINUE
C 95 VELG(I)=GUSTV
C RHG(I)=RHOR
C
C IF(VELG(I).GT.-2.0.AND.IND.GT.0.AND.VELG(I).GT.VELGL)GO TO 101
C VELGL=VELG(I)
C IF(KPRTMH.EQ.0)GO TO 98
C LINES=LINES+1
C IF(LINES.LE.KRCW)GO TO 97
C CALL HEADNG
C WRITE(NOUT,2)
C WRITE(NOUT,1)I,TIMG(I),VELG(I),RHO(I),SLNTRG
C CONTINUE
C T=T+DELTG
C CONTINUE
C 100 CONTINUE
C 101 NTMGST=ICNT+1
C DT1=TIMG(ICNT)-TIMG(ICNT-1)
C DV1=VELG(ICNT)-VELG(ICNT-1)
C TIMG(NTMGST)=TIMG(ICNT-1)-DT1*VELG(ICNT-1)/DV1
C VELG(NTMGST)=0.
C RHO(NTMGST)=RHCA
C IF(KPRTMH.EQ.0)GO TO 102
C WRITE(NOUT,1)ICNT,TIMG(ICNT),VELG(ICNT),RHO(ICNT),SLNTRG

```

GSHSL06
 GSHSL07
 GSHSL08
 GSHSL09
 GSHSL10
 GSHSL11
 GSHSL12
 GSHSL13
 GSHSL14
 GSHSL15
 GSHSL16
 GSHSL17
 GSHSL18
 GSHSL19
 GSHSL20
 GSHSL21
 GSHSL22
 GSHSL23
 GSHSL24
 GSHSL25
 GSHSL26
 GSHSL27
 GSHSL28
 GSHSL29
 GSHSL30
 GSHSL31
 GSHSL32
 GSHSL33
 GSHSL34
 GSHSL35
 GSHSL36
 GSHSL37
 GSHSL38
 GSHSL39
 GSHSL40
 GSHSL41
 GSHSL42
 GSHSL43
 GSHSL44
 GSHSL45
 GSHSL46
 GSHSL47
 GSHSL48
 GSHSL49
 GSHSL50
 GSHSL51
 GSHSL52
 GSHSL53
 GSHSL54
 GSHSL55
 GSHSL56
 GSHSL57

GSTH S158
 GSTH S159
 GSTH S160
 GSTH S161
 GSTH S162
 GSTH S163
 GSTH S164
 GSTH S165
 GSTH S166
 GSTH S167

```

      WRITE (NOUT,1) NIMGST, TIMG(NIMGST), VFLG(NIMGST), RHO(NIMGST)
1     FORMAT(1F,13,2(1X,F8.4),1X,E12.5,1X,F12.5)
2     CONTINUE
      VCGST=VELG(1)
      RETURN
      END
  
```

[illegible]

```

130 IF (Z.GF.11.057506) GO TO 140
140 IF (Z.GT.2.3609977) GO TO 150
150 ASP=0.5
160 GO TO 160
170 ASP=0.0
180 GO TO 160
190 ASP=(-0.332*Z+.388)*Z-.231
200 IF (7.GE.9.5180723) GO TO 170
210 IF (Z.GT.5.0/3.0) GO TO 180
220 ASP=1.0
230 IF (7.GT.1.5589942) GO TO 190
240 GO TO 200
250 ASP=1.67-.011*Z
260 GO TO 190
270 KSP=.88+.072*Z
280 IF (TMTA.GE.TOGUST.AND.TMTA.GE.TORHO) GO TO 200
290 CSP=8.71+.1843*7-104.0/(Z+10.0)
300 CSPG=0.0
310 IF (TMTA.LT.TOGUST) CSPG=CSP
320 CSPR=0.0
330 IF (TMTA.LT.TORHC) CSPR=CSP
340 GO TO 210
350 AGUST=ASP+BCP*7
360 ARHC=AGUST
370 GO TO 220
380 AGUST=ASP+RSP*(1.0+CSPG*TMTA/TOGUST)
390 ARHC=ASP+RSP*(1.0+CSPR*TMTA/TORHC)
400 CSTR=(5.0*Z/SQRT(7.0*(7.0+6.0*Z)))*(1.0-TMTA/TOGUST)*EXP(-AGUST*
410 TMTA/TOGUST)
420 RHOR=(5.0*Z/(7.0+Z))*(1.0-TMTA/TORHC)*EXP(-ARHC*TMTA/TORHC)
430 RETURN
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

```

SUBROUTINE GUSTOR(CR,NTGST,PSIR,PATR,NOR,REST,
2 VELG,TIMC,RHC,OMEGA,PLCTF,PLCTA,
2 AS,AA,AST,AAT,
3 ES,PA,PSI,PAT,TIME,PKL,STALDS,NIMRSP,
4 STRESS,STR,STL,ALLOWS,SYMCOD,ASMCOD,NINTLD,FTG,GFLMRX,GEOM3D)
C
COMMON/Z77/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
COMMON/DISK2/ND2,ITRL2(843),NRECSA,IBUMP,NKD,VOSWS(20)
C
COMMON /XTF/NF(100)
EQUIVALENCE (NF(9),NFREQ)
1 , (NF(2),GAMX)
2 , (NF(43),NRGX), (NF(44),NSBETG)
3 , (NF(11),AMACH), (NF(3),VSS)
4 , (NF(100),SIZECT)
EQUIVALENCE (NF(13),RHQO), (NF(14),SIGMA)
1 , (NF(18),VEL), (NF(19),ALT)
1 , (NF(29),NACCI)
1 , (NF(33),NSTPSS)
3 , (NF(23),KPTRM)
1 , (NF(54),KMAN), (NF(55),AB), (NF(56),AC), (NF(57),INDSYM)
4 , (NF(51),AN)
EQUIVALENCE (NF(61),NORMAX), (NF(62),TIMEMX), (NF(63),FFR)
1 , (NF(54),KGRD), (NF(65),KLPD), (NF(66),HGRD)
2 , (NF(57),KLGAD), (NF(68),NCRTTS)
3 , (NF(89),TAU)
4 , (NF(97),DELT)
5 , (NF(4),PJ)
C
DIMENSION PSTR(1),PATR(1),NOR(1),PFST(1),VELG(1),TIMG(1)
DIMENSION RHQ(1),OMEGA(1),PLCTF(1),PLCTA(1)
DIMENSION AS(1),AA(1),AST(1),AAT(1)
DIMENSION PS(1),PA(1)
DIMENSION PST(1),PAT(1),TIME(1)
DIMENSION PKL(1),STRESS(1),STR(1),STL(1)
DIMENSION ALLOWS(NINTLD,1),STALDS(NINTLD,1)
DIMENSION SYMCOD(1),ASMCOD(1)
DIMENSION CR(3,1)
DIMENSION FTG(2,1)
DIMENSION GEOMBX(1),GEOM90(1)
DIMENSION IGSTABL(20)
DATA ITRMX/5/
DATA EPIHTA/1.0E-02/
TABLE FOR A.S. EQUIVALENCE OF AERO ORIENT TO HLST ORIENT
DATA IGSTAPL/1,2,3,4,5,6,7,8,9,10,11,-11,-10,-9,12,-12,-13,13,0,0/
C
1 FORMAT (6I12)
2 FORMAT (6F12.0)
C

```

```

GUSTOR 2
GUSTOR 3
GUSTOR 4
GUSTOR 5
GUSTOR 6
GUSTOR 7
GUSTOR 8
DISK2 2
GUSTOR10
GUSTDR11
GUSTDR12
GUSTDR13
GUSTDR14
GUSTDR15
GUSTDR16
GUSTDR17
GUSTDR18
GUSTDR19
GUSTDR20
GUSTDR21
GUSTDR22
GUSTDR23
GUSTDR24
GUSTDR25
GUSTDR26
GUSTDR27
GUSTDR28
GUSTDR29
GUSTDR30
GUSTDR31
GUSTDR32
GUSTDR33
GUSTDR34
GUSTDR35
GUSTDR36
GUSTDR37
GUSTDR38
GUSTDR39
GUSTDR40
GUSTDR41
GUSTDR42
GUSTDR43
GUSTDR44
GUSTDR45
GUSTDR46
GUSTDR47
GUSTDR48
GUSTDR49
GUSTDR50
GUSTDR51
GUSTDR52
GUSTDR53

```

```

C      REVISE MAX AND MIN ALLOW LOADS IF RQD
C      IF (KLOAD.EQ.C) GO TO 60
C      DC 50 L=1,NINTLD
C      50 READ(NIN,2)(STALDS(L,J),J=7,8)
C      60 CONTINUE
C      NOW HALVE THOSE C/L ALLOWABLES
C      DC 61 L=1,NINTLD
C      IF (SYMCOD(L).NE.0.ANC.ASMCOD(L).NE.0)GO TO 61
C      STALDS(L,7)=0.50*STALDS(L,7)
C      STALDS(L,8)=0.50*STALDS(L,8)
C      61 CONTINUE
C      INPUT MAX AND MIN STRESSES IF RQD
C      IF (NCRITS.EQ.0)GO TO 70
C      DC 65 L=1,NSTRSS
C      READ(NIN,2)(ALLOW(L,J),J=1,2)
C      IF (SYMCOD(L).NE.0.ANC.ASMCOD(L).NE.0)GO TO 65
C      ALLOW(L,1)=0.50*ALLOW(L,1)
C      ALLOW(L,2)=0.50*ALLOW(L,2)
C      65 CONTINUE
C      70 CONTINUE
C      NOW GET SOLN TIMES
C      IND=0
C      DELTY=DELT
C      TI=-DELT
C      DC 100 I=1,NTMRSP
C      TI=TI+DELT
C      TIME(I)=TI
C      IF (IND.GT.1)GO TO 100
C      IF (TI.LT.0.25)GO TO 100
C      IF (TI.LF.1.0)GO TO 99
C      IND=2
C      TI=TI-DELT
C      DELTY=DELT*2.0
C      TI=TI+DELT
C      TIME(I)=TI
C      GO TO 100
C      99 IF (IND.EQ.1)GO TO 100
C      IND=1
C      TI=TI-DELT
C      DELTY=DELT*5.0

```

GUSTDR54
 GUSTDR55
 GUSTDR56
 GUSTDR57
 GUSTDR58
 GUSTDR59
 GUSTDR60
 GUSTDR61
 GUSTDR62
 GUSTDR63
 GUSTDR64
 GUSTDR65
 GUSTDR66
 GUSTDR67
 GUSTDR68
 GUSTDR69
 GUSTDR70
 GUSTDR71
 GUSTDR72
 GUSTDR73
 GUSTDR74
 GUSTDR75
 GUSTDR76
 GUSTDR77
 GUSTDR78
 GUSTDR79
 GUSTDR80
 GUSTDR81
 GUSTDR82
 GUSTDR83
 GUSTDR84
 GUSTDR85
 GUSTDR86
 GUSTDR87
 GUSTDR88
 GUSTDR89
 GUSTDR90
 GUSTDR91
 GUSTDR92
 GUSTDR93
 GUSTDR94
 GUSTDR95
 GUSTDR96
 GUSTDR97
 GUSTDR98
 GUSTDR99
 GUSTD100
 GUSTD101
 GUSTD102
 GUSTD103
 GUSTD104
 GUSTD105

```

C      TI=TI+DELT
C      TIME(I)=TI
C      CONTINUE
C
C      IF (TIME(NTRSP).GT.TIMEMX) TIME(NTRSP)=TIMEMX
C
C      LOOP THRU DESIRED ORIENTATIONS
C
C      DO 2000 LOOP=1,NORMAX
C      CALL HEADNG
C      NG=NGR(LCOP)
C      SLNTRG=REST(LOOP)
C
C      NG IS BASE ORIENTATION
C
C      CALC INDSYM FOR GUST
C
C      INDSYM=1 IS A SYMMETRIC CASE
C      INDSYM=2 IS AN UNSYMMETRIC CASE
C
C      NGL=IGTARL(NG)
C      NGL IS EQUIV. NG FOR A.S. CASES
C
C      NGLL=IABS(NGL)
C
C      OMEGMX=OMEGA(NFREQ)
C      GAMX=CR(1,NGLL)
C      GAMY=CR(2,NGLL)
C      GAMZ=CR(3,NGLL)
C      IF (NGL.LT.0) GAMY=-GAMY
C      INDSYM=1
C      IF (GAMY.NE.0) INDSYM=2
C
C      LOC=NKD*IRUMP+3
C      NW=6*NBOX
C      CALL READMS(ND2,GFCMBX,NW,LOC)
C      LOC=LOC+1
C      NW=6*NSBFTO
C      CALL READMS(ND2,GECMBD,NW,LOC)
C
C      NOW FIND GEOMETRIC LOC OF BLAST INTERCEPTION ON A/C
C
C      ALT=1.E+10
C
C      DO 30 I1=1,NHGX
C      I2=I1+NBOX
C      I3=I2+NBOX
C      I4=I3+NBOX
C      I5=I4+NBOX
C      I6=I5+NBOX
C      Y1NT=(GECMBX(I1)+GECMBX(I4))/2.0
C      Y1NT=(GECMBX(I2)+GECMBX(I5))/2.0
C
C      GUSTD106
C      GUSTD107
C      GUSTD108
C      GUSTD109
C      GUSTD110
C      GUSTD111
C      GUSTD112
C      GUSTD113
C      GUSTD114
C      GUSTD115
C      GUSTD116
C      GUSTD117
C      GUSTD118
C      GUSTD119
C      GUSTD120
C      GUSTD121
C      GUSTD122
C      GUSTD123
C      GUSTD124
C      GUSTD125
C      GUSTD126
C      GUSTD127
C      GUSTD128
C      GUSTD129
C      GUSTD130
C      GUSTD131
C      GUSTD132
C      GUSTD133
C      GUSTD134
C      GUSTD135
C      GUSTD136
C      GUSTD137
C      GUSTD138
C      GUSTD139
C      GUSTD140
C      GUSTD141
C      GUSTD142
C      GUSTD143
C      GUSTD144
C      GUSTD145
C      GUSTD146
C      GUSTD147
C      GUSTD148
C      GUSTD149
C      GUSTD150
C      GUSTD151
C      GUSTD152
C      GUSTD153
C      GUSTD154
C      GUSTD155
C      GUSTD156
C      GUSTD157

```

GUSID158
GUSID159
GUSID160
GUSID161
GUSID162
GUSID163
GUSID164
GUSID165
GUSID166
GUSID167
GUSID168
GUSID169
GUSID170
GUSID171
GUSID172
GUSID173
GUSID174
GUSID175
GUSID176
GUSID177
GUSID178
GUSID179
GUSID180
GUSID181
GUSID182
GUSID183
GUSID184
GUSID185
GUSID186
GUSID187
GUSID188
GUSID189
GUSID190
GUSID191
GUSID192
GUSID193
GUSID194
GUSID195
GUSID196
GUSID197
GUSID198
GUSID199
GUSID200
GUSID201
GUSID202
GUSID203
GUSID204
GUSID205
GUSID206
GUSID207
GUSID208
GUSID209

```

ZINT=(GEOMBX(I3)+GEOMBX(I6))/2.0
YINTL=-YINT
DY=GEOMRX(I5)-GEOMBX(I2)
DZ=GEOMBX(I6)-GEOMBX(I3)
DCL=SQRT(DY**2+DZ**2)
SINDIH=DZ/DCL
COSDIH=DY/DCL
THETAR=GAMZ*COSDIH-GAMY*SINDIH
THETAL=GAMZ*COSDIH+GAMY*SINDIH
IF(ABS(THETAR).LT.EPHTA)GO TO 25
ALR=GAMX*XINT+GAMY*YINT+GAMZ*ZINT
IF(ALR.GT.ALI)GO TO 25
ALI=ALR
X=XINT
Y=YINT
Z=ZINT
25  CONTINUE
IF(ABS(THETAL).LT.EPHTA)GO TO 30
ALL=GAMX*XINT+GAMY*YINTL+GAMZ*ZINT
IF(ALL.GT.ALI)GO TO 30
ALI=ALL
X=XINT
Y=YINTL
Z=ZINT
30  CONTINUE
DO 40 I1=1,NSBETO
I2=I1+NSBETO
I3=I2+NSBETO
I4=I3+NSBETO
I5=I4+NSBETO
I6=I5+NSBETO
XINT=(GEOMBD(I1)+GEOMBD(I4))/2.0
YINT=(GEOMBD(I2)+GEOMBD(I5))/2.0
ZINT=(GEOMBD(I3)+GEOMBD(I6))/2.0
YINTL=-YINT
ALP=GAMX*XINT+GAMY*YINT+GAMZ*ZINT
IF(ALP.GT.ALI)GO TO 35
ALI=ALP
X=XINT
Y=YINT
Z=ZINT
35  CONTINUE
ALI=GAMX*XINT+GAMY*YINTL+GAMZ*ZINT
IF(ALL.GT.ALI)GO TO 40
ALI=ALL
X=XINT
Y=YINTL
Z=ZINT
40  CONTINUE
NOW THE TIME OF BURST INTERCEPT (WRT ORIGIN OF AAS) IS

```


GUSTD262
GUSTD263
GUSTD264
GUSTD265
GUSTD266
GUSTD267
GUSTD268
GUSTD269
GUSTD270
GUSTD271
GUSTD272
GUSTD273
GUSTD274
GUSTD275
GUSTD276
GUSTD277
GUSTD278
GUSTD279
GUSTD280
GUSTD281
GUSTD282
GUSTD283
GUSTD284
GUSTD285
GUSTD286
GUSTD287
GUSTD288
GUSTD289
GUSTD290
GUSTD291
GUSTD292
GUSTD293
GUSTD294
GUSTD295
GUSTD296
GUSTD297
GUSTD298
GUSTD299
GUSTD300
GUSTD301
GUSTD302
GUSTD303
GUSTD304
GUSTD305
GUSTD306
GUSTD307
GUSTD308
GUSTD309
GUSTD310
GUSTD311
GUSTD312
GUSTD313

```

CALL LAUNCH(TIME,ITERS,
1 NINTLD,NTMRSP,NG,NGL,GAMX,GAMY,GAMZ,INDSYM,PKL,PST,PAT,
1 PSTP,PATP,STALDS,TIME,VGUST,VGUSTR,PO,DELPB,DELPBP,VSS)
IF (NSTRSS.NF.0)CALL STRSCH
1 ITERS,
1 NINTLD,NTMRSP,PKL,PST,PAT,STL,STR,ALLOW,S,TIME,VGLST,VGUSTR,PO,
1 DELPR,DELPBR,VSS,STRESS,NSTRSS,NCRITS)
FIND NEW RANGE FOR ITERATION
CALL RANGE (DELPBR,RTA)
PR=1.0
IF (RTA.GT.0.55)RR=1.0+0.06*(RTA-0.55)**(1./3.)
RTA=RTA/RR
SLNTRG=RTA/SCALR
TARR=0.
CALL TAR (RTA,TARR)
TARR=-TARR*SCALT
CALL FLTPCS(TARR,SLNTRG,VEL,ALT,AN,AB,AC,GAMX,GAMY,GAMZ,KMAN,
1 QUM,XH,YR,ZR,L,XE,YE,ZE)
IF (K1PT.EQ.0)GO TO 1000
IF (ABS(1.0-VGUST/VGUSTP).LT.0.025)ITERS=1
IF (ITERS.NF.0)GO TO 1000
ITPR=ITPR+1
GO TO 500
C 1000 CONTINUE
IF (ITERS.NF.0)WRITE(NGUT,4)ITPR,SLNTRG
4 FORMAT(1H0,20X,CONVERGED SOLUTION /1H,
1 20X,I2,ITERATIONS /1H,
1 20X,CRITICAL RANGE IS ,F10.2, FT. )
C
CALL HEADING
WRITE(NGUT,8)
DC 700 I=1,NINTLD
LINES=LINES+1
IF (LINES.LT.KRCW)GO TO 690
CALL HEADING
WRITE(NGUT,8)
690 CONTINUE
700 WRITE(NGUT,9)I,(STALDS(I,J),J=1,6)
9 FORMAT(1H,14,3(1X,F4.1),3(1X,F12.5))
8 FORMAT(1H0,20X,INTEGRATED LOAD DEFINITIONS /1H0,
1 LOAD BEAM CODE COMP ,9X,XAAS ,9X,ZAAS )
C 2000 CONTINUE
C

```

GUSID314
GUSID315

RETURN
END


```

54 IFI
55 IFI
56 IFI
57 IFI
58 IFI
59 IFI
60 IFI
61 IFI
62 IFI
63 IFI
64 IFI
65 IFI
66 IFI
67 IFI
68 IFI
69 IFI
70 IFI
71 IFI
72 IFI
73 IFI
74 IFI
75 IFI
76 IFI
77 IFI
78 IFI
79 IFI
80 IFI
81 IFI

```

```

      BRI = DEL*(AI2-AI1)
      G(J,K) = G(J,K) + TWOOP1*DEN*(AI2*C2-AI1*C1-BRI*DEN*(S2-S1))
100 CONTINUE
      C1=C2
      S1=S2
110 CONTINUE
120 CONTINUE
      C
      C
      C      NOW FINISH TAIL INTEGRATION FOR FREQS ABOVE LAST INPUT FREQ
      DEL=1.0/(FINREQ-C*WEGA(NFREQ))
      DO 150 K=2,NTP
      T=TIME(K)
      DEN=1.0/T
      ANG1=C*WEGA(NFREQ)*T
      C1=COS(ANG1)
      S1=SIN(ANG1)
      ANG2=FINREQ*T
      C2=CCS(ANG2)
      S2=SIN(ANG2)
      DO 150 J=1,NMP
      AI1 = A(2,J,NFREQ)
      BRI = -A(2,J,NFREQ)*DEL
      G(J,K) = G(J,K) + TWOOP1*DEN*(-AI1*C1-BRI*DEN*(S2-S1))
150 CONTINUE
      C
      RETURN
      END

```



```

70      F22=D72*F(I221)-D71*F(I222)
      GO TO 80
      I11=(IX-1)*NY+IY
      I12=I11+1
      I21=I11+NY
      I22=I21+1
      F11=F(I11)
      F12=F(I12)
      F21=F(I21)
      F22=F(I22)
      DY=VT(IY+1)-VT(IY)
      DY1=(VT(IY)-Y)/DY
      DY2=(VT(IY+1)-Y)/DY
      F1=DY2*F11-DY1*F12
      F2=DY2*F21-DY1*F22
      GO TO 100
      F1=F(I1)
      F2=F(I1+1)
      DX=XT(IX+1)-XT(IX)
      DX1=(XT(IX)-X)/DX
      DX2=(XT(IX+1)-X)/DX
      FF=DX2*F1-DX1*F2
      RETURN
110     WRITE (5,150) X,XT(I),I=1,NX)
      GO TO 140
120     WRITE (5,160) Y,VT(I),I=1,NY)
      GO TO 140
130     WRITE (5,170) Z,VT(I),I=1,NZ)
140     STOP
      C
      C
150     FORMAT (38H FIRST ARGUMENT IS OUTSIDE TABLE, X =,E13.6/5H XT =,/(
160     11X,5E13.6))
      FORMAT (39H SECOND ARGUMENT IS OUTSIDE TABLE, Y =,E13.6/5H VT =,/(
170     11X,5E13.6))
      FORMAT (38H THIRD ARGUMENT IS OUTSIDE TABLE, Z =,E13.6/5H ZT =,/(
      11X,5E13.6))
      END

```

```

INTRP854
INTRP855
INTRP856
INTRP857
INTRP858
INTRP859
INTRP860
INTRP861
INTRP862
INTRP863
INTRP864
INTRP865
INTRP866
INTRP867
INTRP868
INTRP869
INTRP870
INTRP871
INTRP872
INTRP873
INTRP874
INTRP875
INTRP876
INTRP877
INTRP878
INTRP879
INTRP880
INTRP881
INTRP882
INTRP883
INTRP884
INTRP885
INTRP886
INTRP887
INTRP888
INTRP889
INTRP890
INTRP891

```

AD-A106 520

DOUGLAS AIRCRAFT CO. LONG BEACH CA
NUCLEAR BLAST RESPONSE COMPUTER PROGRAM. VOLUME III. PROGRAM LI--ETC(U)
AUG 81 J A MCOREV, H H CROXEN, T P KALMAN DAA001-75-C-0216

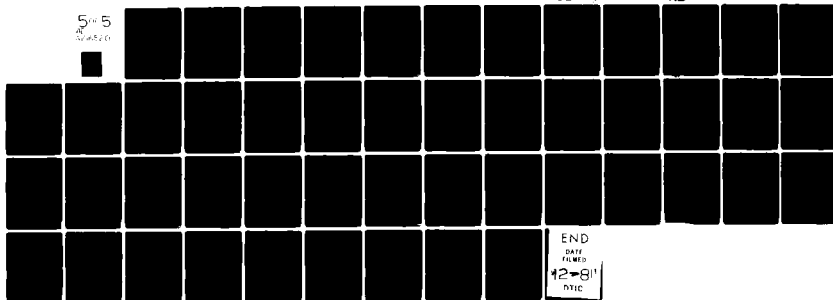
F/8 18/3

UNCLASSIFIED

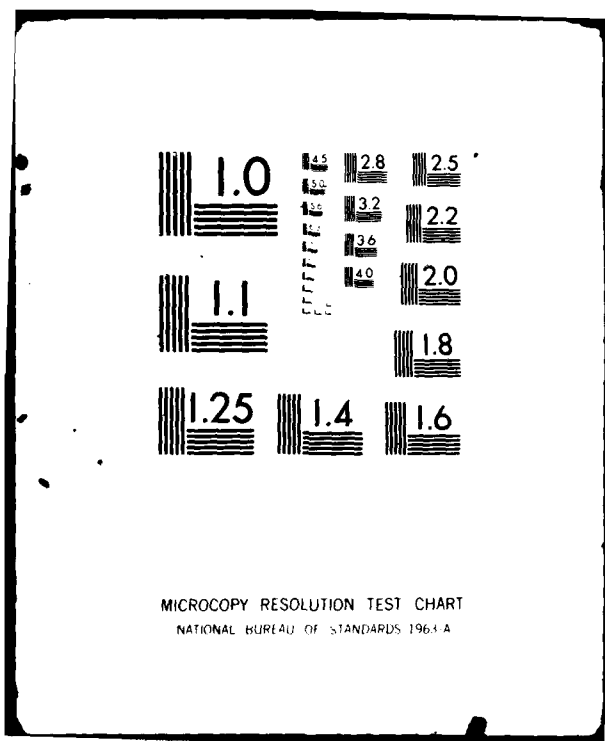
AFWL-TR-81-32-VOL-3

NL

5-5
000000



END
DATE
FILMED
12-81
DTIC



```

SUBROUTINE LOADCH(PLOT,ITERS,GAMY,GAMZ,INDSYM,
1 NINTLD,NTMRSP,NGL,NGLL,GAMX,GAMZ,INDSYM,
1 PKL,PST,PAT,PSTR,PATR,STALDS,TIME,
1 VGUST,VGUSTR,PO,DELPR,DELPRR,VSS)
COMMON/XTF/NF(100)
EQUIVALENCE (NF(25),KPRBLS)
1 , (NF(28),IPLRL)
COMMON/ZZ/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
DIMENSION PKL(NINTLD,8)
DIMENSION PST(NINTLD,1),PAT(NINTLD,1),PSTR(1),PATR(1)
DIMENSION STALDS(NINTLD,8),TIME(1)
DIMENSION PLOT(1)
1 FORMAT(IHO,20X,LOAD TIME HISTORIES /IHO,
1 20X,ORIENTATION NO.,13,GAMX = ,E12.5,GAMY = ,E12.5,
1 GAMZ = ,E12.5)
2 FORMAT(IH,14,2X,F8.4,8(1X,E12.5))
3 FORMAT(IHO,NO,6X,TIME,4(5X,R-STA,13,5X,L-STA,13))
4 FORMAT(IHO,20X,MAXIMUM LOADS /IHO,STA,6X,TIME,
1 6X,R SIDE+,5X,TIME,6X,R SIDE-,5X,TIME,6X,L SIDE+,
1 5X,TIME,6X,L SIDE-,8X,MAX+,8X,MAX-)
5 FORMAT(IH,14,1X,4(1X,F8.4,1X,E12.5),2(1X,E12.5))
6 FORMAT(IHO,20X,ORIENTATION NO.,14)
11 FORMAT(IHO,20X,LOAD TIME HISTORY FOR RHS LOAD NO.,14,
1 ORIENTATION NO.,14)
12 FORMAT(IHO,20X,LOAD TIME HISTORY FOR LHS LOAD NO.,14,
1 ORIENTATION NO.,14)
7 FORMAT(IHO,20X,MATERIAL VELOCITY=,F10.4,FPS /IH,PATR,
1 20X,MAY ALLOWABLE VELOCITY=,F10.4,FPS /IH,20X,P AMBIENT=,
1 F10.4,PSI /IH,20X,OVERPRESSURE=,F10.4,PSI /IH,20X,
1 MAX ALLOW. OVERPRESSURE=,F10.4,PSI)
ITERS=0
HERE FIND PROPER DIRECTIONS FOR INCREMENTAL LOADS (TRIM DEPENDENT)
AMSE=1.0
AMAE=1.0
IF (NGLL.LT.0) AMA=-1.0
TRIM LOADS ARE IN PSTR,PATR
INCREMENTAL LOADS ARE IN PST,PAT
TOTAL LOADS ARE IN PL,PA (LHS, RHS)
CC 500 I=1,NINTLD
CC 490 J=1,8
490 PKL(I,J)=0.
IF (INDSYM.EQ.1) GO TO 300

```

LOADCH 2
LOADCH 3
LOADCH 4
LOADCH 5
LOADCH 6
LOADCH 7
LOADCH 8
LOADCH 9
LOADCH 10
LOADCH 11
LOADCH 12
LOADCH 13
LOADCH 14
LOADCH 15
LOADCH 16
LOADCH 17
LOADCH 18
LOADCH 19
LOADCH 20
LOADCH 21
LOADCH 22
LOADCH 23
LOADCH 24
LOADCH 25
LOADCH 26
LOADCH 27
LOADCH 28
LOADCH 29
LOADCH 30
LOADCH 31
LOADCH 32
LOADCH 33
LOADCH 34
LOADCH 35
LOADCH 36
LOADCH 37
LOADCH 38
LOADCH 39
LOADCH 40
LOADCH 41
LOADCH 42
LOADCH 43
LOADCH 44
LOADCH 45
LOADCH 46
LOADCH 47
LOADCH 48
LOADCH 49
LOADCH 50
LOADCH 51
LOADCH 52
LOADCH 53

LOADCH54
LOADCH55
LOADCH56
LOADCH57
LOADCH58
LOADCH59
LOADCH60
LOADCH61
LOADCH62
LOADCH63
LOADCH64
LOADCH65
LOADCH66
LOADCH67
LOADCH68
LOADCH69
LOADCH70
LOADCH71
LOADCH72
LOADCH73
LOADCH74
LOADCH75
LOADCH76
LOADCH77
LOADCH78
LOADCH79
LOADCH80
LOADCH81
LOADCH82
LOADCH83
LOADCH84
LOADCH85
LOADCH86
LOADCH87
LOADCH88
LOADCH89
LOADCH90
LOADCH91
LOADCH92
LOADCH93
LOADCH94
LOADCH95
LOADCH96
LOADCH97
LOADCH98
LOADCH99
LOADC100
LOADC101
LOADC102
LOADC103
LOADC104
LOADC105

```

C      CALC LHS RHS LOADS (UNSYMMETRIC LOAD CONDITION)
C      DO 200 J=1, NIMRSP
C      IF (PSTR(I)-PATR(I)+AMS*PST(I,J)-AMA*PAT(I,J)
C      PST(I,J)=PSTR(I)+PATR(I)+AMS*PST(I,J)+AMA*PAT(I,J)
C      PAT(I,J)=TEMP
C      FIND PEAK LHS AND RHS + AND - LOADS
C      IF (PST(I,J).LT.PKL(I,2)) GO TO 201
C      PKL(I,2)=PST(I,J)
C      PKL(I,1)=TIME(J)
C      GO TO 202
C      201 IF (PST(I,J).GT.PKL(I,4)) GO TO 202
C      PKL(I,4)=PST(I,J)
C      PKL(I,3)=TIME(J)
C      202 IF (PAT(I,J).LT.PKL(I,6)) GO TO 203
C      PKL(I,6)=PAT(I,J)
C      PKL(I,5)=TIME(J)
C      GO TO 204
C      203 IF (PAT(I,J).GT.PK (I,8)) GO TO 204
C      PKL(I,8)=PAT(I,J)
C      PKL(I,7)=TIME(J)
C      204 CONTINUE
C      200 CONTINUE
C      GO TO 495
C      300 CONTINUE
C      CALC LHS RHS LOADS (SY-METRIC CONDITION)
C      DO 350 J=1, NIMRSP
C      IF (PSTR(I)+AMS*PST(I,J)-PATR(I)
C      PST(I,J)=PSTR(I)+AMS*PST(I,J)+PATR(I)
C      PAT(I,J)=TEMP
C      FIND PEAK LHS AND RHS + AND - LOADS
C      IF (PST(I,J).LT.PKL(I,2)) GO TO 301
C      PKL(I,2)=PST(I,J)
C      PKL(I,1)=TIME(J)
C      GO TO 302
C      301 IF (PST(I,J).GT.PKL(I,4)) GO TO 302
C      PKL(I,4)=PST(I,J)
C      PKL(I,3)=TIME(J)
C      302 IF (PAT(I,J).LT.PKL(I,6)) GO TO 303
C      PKL(I,6)=PAT(I,J)
C      PKL(I,5)=TIME(J)
C      GO TO 304
C      303 IF (PAT(I,J).GT.PKL(I,8)) GO TO 304
C      PKL(I,8)=PAT(I,J)
C      PKL(I,7)=TIME(J)
C      304 CONTINUE
C      350 CONTINUE
C

```

LOADC106
LOADC107
LOADC108
LOADC109
LOADC110
LOADC111
LOADC112
LOADC113
LOADC114
LOADC115
LOADC116
LOADC117
LOADC118
LOADC119
LOADC120
LOADC121
LOADC122
LOADC123
LOADC124
LOADC125
LOADC126
LOADC127
LOADC128
LOADC129
LOADC130
LOADC131
LOADC132
LOADC133
LOADC134
LOADC135
LOADC136
LOADC137
LOADC138
LOADC139
LOADC140
LOADC141
LOADC142
LOADC143
LOADC144
LOADC145
LOADC146
LOADC147
LOADC148
LOADC149
LOADC150
LOADC151
LOADC152
LOADC153
LOADC154
LOADC155
LOADC156
LOADC157

```

495 CONTINUE
500 CONTINUE
C
C
IF (KPRBL.EQ.0) GO TO 600
CALL HEADNG
WRITE (NCUT,1) NGL,GAMX,GAMY,GAMZ
JF=0
LINE5=4
C
550 JS=JF+1
JF=JS+3
IF (JF.GT.NINTLD) JF=NINTLD
WRITE (NCUT,3) (J,J,J=JS,JF)
NC 520 I=1,NTMRSP
LINE5=LINE5+1
IF (LINE5.LE.KPCW) GO TO 515
CALL HEADNG
WRITE (NCUT,1) NGL,GAMX,GAMY,GAMZ
WRITE (NCUT,3) (J,J,J=JS,JF)
LINE5=LINE5+4
C
515 CF NUF
520 WF NCUT,2) I,TIME(I), (PST(J,I),PAT(J,I),J=JS,JF)
F GE.NINTLD) GO TO 600
GO TO 550
600 CONTINUE
C
IF (IPLBL.EQ.0) GO TO 630
NEW PLOT LOAD TIME HIST (RHS)
NC 620 I=1,NINTLD
IF (IPLBL.EQ.2) GO TO 625
NC 610 J=1,NTMRSP
PLOT(J)=PST(I,J)
CALL PLT1(NTMRSP,TIME,PLOT,I)
WRITE (NCUT,1) I,NGL
C
625 CONTINUE
IF (IPLBL.EQ.1) GO TO 620
IF (IPLBL.EQ.3.AND.INDSYM.EQ.1) GO TO 620
C
NC 626 J=1,NTMRSP
PLOT(J)=PAT(I,J)
CALL PLT1(NTMRSP,TIME,PLOT,I)
WRITE (NCUT,12) I,NGL
C
620 CONTINUE
C
630 CONTINUE
C

```

LOADC158
LOADC159
LOADC160
LOADC161
LOADC162
LOADC163
LOADC164
LOADC165
LOADC166
LOADC167
LOADC168
LOADC169
LOADC170
LOADC171
LOADC172
LOADC173
LOADC174
LOADC175
LOADC176
LOADC177
LOADC178
LOADC179
LOADC180
LOADC181
LOADC182
LOADC183
LOADC184
LOADC185
LOADC186
LOADC187
LOADC188
LOADC189
LOADC190
LOADC191
LOADC192
LOADC193
LOADC194
LOADC195
LOADC196
LOADC197
LOADC198
LOADC199
LOADC200
LOADC201
LOADC202
LOADC203
LOADC204
LOADC205
LOADC206
LOADC207
LOADC208
LOADC209

```

C
CALL HEADNG
WRITE (NOUT,4)
PKP=0.
PKM=0.
IIP=0.
IIM=0.
CC 650 I=1,NINTLD
LINES=LINES+1
IF (LINES.LT.KRCW) GO TO 642
CALL HEADNG
WRITE (NOUT,4)
CONTINUE
642 WRITE (NOUT,5) I, (PKL(I,J),J=1,8), (STALDS(I,J),J=7,8)
FIND MAX POS LOAD RATIO
DO 640 J=2,6,4
IF (STALDS(I,7).EQ.0) GO TO 640
RATIO=PKL(I,J)/STALDS(I,7)
IF (RATIO.LT.PKM) GO TO 640
PKP=RATIO
IIP=I
IIM=I
TIMP=PKL(I,J-1)
CONTINUE
640 FIND MAX NEG LOAD RATIO
DO 645 J=4,8,4
IF (STALDS(I,8).EQ.0) GO TO 645
RATIO=PKL(I,J)/STALDS(I,8)
IF (RATIO.LT.PKM) GO TO 645
PKM=RATIO
IIM=I
TIMM=PKL(I,J-1)
CONTINUE
645 CONTINUE
650 CONTINUE
C
FIND NEW ALLOW GUST VFL
PKPR=1.0
PKMR=1.0
IF (IIP.FQ.0) GO TO 660
PKPR=PKP
660 IF (IIM.FQ.0) GO TO 670
PKMR=PKM
CONTINUE
670 PKMAX = A*AXL(PKPR,PKMR)
IF (ABS(1.-PKMAX).LE.0.025) ITERS=1
VGUSTR=VGUST/PKMAX
DELPP=PO*DELPPR
RAT=VGUSTR/VSS
FCT=(21./25.)*RAT**2
DELPPR=FCT+SQR(FCT**2+(49./25.)*RAT**2)
WRITE (NOUT,6) NGL

```

LOADC210
LOADC211
LOADC212
LOADC213
LOADC214
LOADC215
LOADC216
LOADC217
LOADC218

WRITE (NOUT,7) VGUST, VGUSTR, PO, CELP, DELPP
WRITE (NOUT,10) IIP, PKPR, TIMP, IIM, PKMR, TIMM
FCRMA T I IHO, 20X, MAXIMUM POS. AND NEG. LOAD RADIUS /IHO,
1 20X, STA, 4X, LOAD, 5X, TIME /IH, F8.4))
1 20X, I3, 2(IX, F8.4) /IH, 20X, I3, 2(IX, F8.4))

RETURN
END

C


```

SUBROUTINE RANGE(P,R)
COMMON/Z77/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
GIVEN OVERPRESSURE RATIO P, SCALED RANGE R IS RETURNED
DIMENSION PT(7),A(7),B(7),C(7)
DATA PT/0.12957,0.28257,0.49058,0.73087,1.25051,
1 3.01555,6.03899/
DATA A/4.531507,2.636265,1.662156,1.198027,0.915294,
1 0.633333,0.451231/
DATA B/67.394528,9.601168,1.777405,0.496103,0.165727,
1 0.030217,0.006696/
DATA C/-24.723682,-8.132669,-3.196272,-1.639432,
1 -0.945597,-0.437786,-0.221178/

IF(P.GT.0.09697)GO TC 20
R=(0.37675/P)*0.809454
GO TC 100

20 DC 30 I=1,7
IF(P.LE.PI(I))GO TC 50
30 CONTINUE

IF(P.GT.69.9195)GO TC 40
R=(0.27889/P)*0.391512
GO TC 100

40 WRITE(NOUT,60)P
60 FORMAT(1H1,20X, OVERPRESSURE RATIO OF .F12.5,
1 IS GREATER THAN MAX ALLOWABLE )
STOP

50 R=A(I)-SQRT(A(I)**2+B(I)*P+C(I))
100 CONTINUE
RETURN
END

```

RANGE 2
 RANGE 3
 RANGE 4
 RANGE 5
 RANGE 6
 RANGE 7
 RANGE 8
 RANGE 9
 RANGE 10
 RANGE 11
 RANGE 12
 RANGE 13
 RANGE 14
 RANGE 15
 RANGE 16
 RANGE 17
 RANGE 18
 RANGE 19
 RANGE 20
 RANGE 21
 RANGE 22
 RANGE 23
 RANGE 24
 RANGE 25
 RANGE 26
 RANGE 27
 RANGE 28
 RANGE 29
 RANGE 30
 RANGE 31
 RANGE 32
 RANGE 33
 RANGE 34
 RANGE 35
 RANGE 36
 RANGE 37
 RANGE 38
 RANGE 39
 RANGE 40
 RANGE 41
 RANGE 42
 RANGE 43
 RANGE 44
 RANGE 45

STRSCH1
STRSCH2
STRSCH3
STRSCH4
STRSCH5
STRSCH6
STRSCH7
STRSCH8
STRSCH9
STRSCH10
STRSCH11
STRSCH12
STRSCH13
STRSCH14
STRSCH15
STRSCH16
STRSCH17
STRSCH18
STRSCH19
STRSCH20
STRSCH21
STRSCH22
STRSCH23
STRSCH24
STRSCH25
STRSCH26
STRSCH27
STRSCH28
STRSCH29
STRSCH30
STRSCH31
STRSCH32
STRSCH33
STRSCH34
STRSCH35
STRSCH36
STRSCH37
STRSCH38
STRSCH39
STRSCH40
STRSCH41
STRSCH42
STRSCH43
STRSCH44
STRSCH45
STRSCH46
STRSCH47
STRSCH48
STRSCH49
STRSCH50
STRSCH51
STRSCH52
STRSCH53

```

SUBROUTINE STRSCH1
  1 ITERS,NINTLD,NTMRSP,
  1 PKL,PST,PAT,STL,STR,ALLOW,TIME,
  1 VGUST,VGUSTR,PO,DELPR,DELPAR,VSS,
  1 STRESS,NSTRSS,NCRITS)
  C
  COMMON/777/CASE(48),NIN,NDUT,KROW,LINES,IPRNT,NER
  C
  COMMON/XTF/NF(100)
  EQUIVALENCE (NF(25),KPRBLS)
  C
  DIMENSION PKL(NSTRSS,8)
  DIMENSION PST(NINTLD,1),PAT(NINTLD,1)
  DIMENSION ALLOW(NSTRSS,2),TIME(1)
  DIMENSION STL(NSTRSS,1),STR(NSTRSS,1)
  DIMENSION STRESS(NSTRSS,1)
  C
  1 FORMAT(1H0,20X, STRESS TIME HISTORIES )
  2 FCPMAT(1H,14,2X,F8.4,8(1X,E12.5))
  3 FCPMAT(1H0, NO,6X, TIME,4(5X, R-STA, I3,5X, L-STA, I3))
  4 FCPMAT(1H0,20X, MAXIMUM STRESSES /1H0, STA,6X, TIME
  1 6X, R SIDE+,5X, TIME,6X, R SIDE+,5X, TIME,6X, L SIDE+,
  1 5X, TIME,6X, L SIDE-,8X, MAX+,8X, MAX- )
  5 FCPMAT(1H,14,1X,4(1X,F8.4,1X,E12.5),2(1X,E12.5))
  7 FCPMAT(1H0,20X, MATERIAL VELOCITY=,F10.4, FPS /1H, P
  1 20X, MAX ALLOWABLE VELOCITY=,F10.4, FPS /1H, PSI /1H,20X,
  1 F10.4, PSI /1H,20X, OVERPRESSURE=,F10.4, PSI /1H,20X,
  1 MAX ALLOW. OVERPRESSURE=,F10.4, PSI )
  C
  IF (NCRITS.NE.0) ITERS=0
  C
  DO 500 I=1,NSTRSS
  DO 490 J=1,8
  490 PKL(I,J)=0.
  C
  DO 200 J=1,NTMRSP
  STL(I,J)=0.
  STR(I,J)=0.
  C
  DO 150 K=1,NINTLD
  STL(I,J)=STL(I,J)+STRESS(I,K)*PAT(K,J)
  STR(I,J)=STR(I,J)+STRESS(I,K)*PST(K,J)
  150
  FIND PEAK LHS AND RHS + AND - STRESS
  IF (STR(I,J).LT.PKL(I,2)) GC TO 201
  PKL(I,2)=STR(I,J)
  PKL(I,1)=TIME(J)
  GC TO 202
  201 IF (STR(I,J).GT.PKL(I,4)) GC TO 202
  PKL(I,4)=STR(I,J)

```

STR SCH54
STR SCH55
STR SCH56
STR SCH57
STR SCH58
STR SCH59
STR SCH60
STR SCH61
STR SCH62
STR SCH63
STR SCH64
STR SCH65
STR SCH66
STR SCH67
STR SCH68
STR SCH69
STR SCH70
STR SCH71
STR SCH72
STR SCH73
STR SCH74
STR SCH75
STR SCH76
STR SCH77
STR SCH78
STR SCH79
STR SCH80
STR SCH81
STR SCH82
STR SCH83
STR SCH84
STR SCH85
STR SCH86
STR SCH87
STR SCH88
STR SCH89
STR SCH90
STR SCH91
STR SCH92
STR SCH93
STR SCH94
STR SCH95
STR SCH96
STR SCH97
STR SCH98
STR SCH99
STR SCI00
STR SCI01
STR SCI02
STR SCI03
STR SCI04
STR SCI05

```

202 PKL(I,3)=TIME(J)
   IF (STL(I,J).LT.PKL(I,6)) GO TO 203
   PKL(I,6)=STL(I,J)
   PKL(I,5)=TIME(J)
   GO TO 204
203 IF (STL(I,J).GT.PKL(I,8)) GO TO 204
   PKL(I,8)=STL(I,J)
   PKL(I,7)=TIME(J)
204 CONTINUE
200 CONTINUE
   C
   C 500 CONTINUE
   C
   C
   IF (KPBLS.EQ.0) GO TO 390
   CALL HEADNG
   WRITE (NOUT,1)
   JF=0
   LINES=2
   C
   C 550 JS=JF+1
   JF=JS+3
   IF (JF.GT.NSTRSS) JF=NSTRSS
   WRITE (NOUT,3) (J,J,J=JS,JF)
   DO 520 I=1,NTMRSP
   LINES=LINES+1
   IF (LINES.LE.KROW) GO TO 300
   CALL HEADNG
   WRITE (NOUT,1)
   WRITE (NOUT,3) (J,J,J=JS,JF)
   LINES=LINES+2
   CONTINUE
300 WRITE (NOUT,2) I,TIME(I),(STR(J,I),STL(J,I),J=JS,JF)
520 IF (JF.GE.NSTRSS) GO TO 600
   GO TO 550
600 CONTINUE
   C
   C 390 CONTINUE
   C
   CALL HEADNG
   WRITE (NOUT,4)
   IIP=0
   IIM=0
   PKP=0.
   PKV=0.
   DO 650 I=1,NSTRSS
   WRITE (NOUT,5) I,(PKL(I,J),J=1,8)
   DO 640 J=2,6,4
   IF (ALLOWS(I,1).EQ.0) GO TO 640
   RATIO=PKL(I,J)/ALLOWS(I,1)
   IF (RATIO.LT.PKP) GO TO 640
   PKP=RATIO

```

```

640      IIP=I
        CONTINUE
        DO 645 J=4,8,4
          IF (ALLOWS(I,2).EQ.0) GO TO 645
          RATIO=PKL(I,J)/ALLOWS(I,2)
          IF (RATIO.LT.PKM) GO TO 645
          PKM=RATIO
          IIM=I
        645 CONTINUE
        C
        C 650 CONTINUE
        C
        C IF (NCRITS.EQ.0) GO TO 1000
        C FIND NEW ALLOW GUST VEL
        C
        C PKPR=1.0
        C PKMR=1.0
        C IF (IIP.EQ.0) GO TO 660
        C PKPR=PKP
        C IF (IIM.EQ.0) GO TO 670
        C PKMR=PKM
        C 660 CONTINUE
        C PKMAX=AMAX1(PKPR,PKMR)
        C IF (ABS(1.-PKMAX).LE.0.025) ITERS=1
        C VGUSTR=VGUST*PKMAX
        C DELP=PO*DELP
        C RAT=VGUSTR/VSS
        C FCT=(21./25.)*RAT**2
        C DELPRR=FCT+SQRT(FCT**2+(49./25.)*RAT**2)
        C DELPP=PO*DELPRR
        C WRITE (NOUT,7) VGUST,VGUSTR,PO,DELP,DELPP
        C 1000 CONTINUE
        C
        C RETURN
        C END

```

```

STR SCI107
STR SCI108
STR SCI109
STR SCI110
STR SCI111
STR SCI112
STR SCI113
STR SCI114
STR SCI115
STR SCI116
STR SCI117
STR SCI118
STR SCI119
STR SCI120
STR SCI121
STR SCI122
STR SCI123
STR SCI124
STR SCI125
STR SCI126
STR SCI127
STR SCI128
STR SCI129
STR SCI130
STR SCI131
STR SCI132
STR SCI133
STR SCI134
STR SCI135
STR SCI136
STR SCI137
STR SCI138
STR SCI139
STR SCI140
STR SCI141

```

```

      SUBROUTINE TAR(R,T)
      GIVEN SCALED RANGE, TIME OF ARRIVAL OF THE SHOCK WAVE IN T.
      RETURNS THE SCALED TIME OF ARRIVAL IN T, WITH R=0, SUBROUTINE TAR
      GIVEN SCALED TIME OF ARRIVAL IN T, WITH R=0, SUBROUTINE TAR
      RETURNS THE SCALED SHOCK RADIUS IN R.
      DIMENSION RS(5), A(5), B(5), C(5)
      DATA PS/5.867375, 1.68, 0.43, 0.265, 0.115/, A/-632144, -.3863754, -.141
      1976, -0.0162814, -0.0031878, 0.895889, 0.810069, 0.5227985, 0.0349106, -
      20.04505377, 0.0, 0.00731329, 0.0938408, 0.548666, 0.663855/
      IF (C.EQ.0.0) GO TO 40
      DO 10 I=1,5
      IF (R.GE.RS(I)) GO TO 20
      CONTINUE
      WRITE (5,70) R
      STOP
      T=(C(I)*R+R(I))*R+A(I)
      RETURN
      R=(T-A(I))/B(I) GO TO 30
      IF (R.GE.PS(I)) GO TO 30
      DO 50 I=2,5
      TT=(C(I)*PS(I)+R(I))*RS(I)+A(I)
      IF (T-TT) GO TO 50
      R=(-R(I)+SQRT(B(I)**2+4.0*C(I)*(T-A(I))))/(2.0*C(I))
      GO TO 30
      CONTINUE
      WRITE (6,50) T
      STOP
      FORMAT (15H SCALED TIME OF ,F11.4,49H IS LESS THAN MINIMUM PERMISSI
      11PLE VALUE, 0.000418)
      FORMAT (17H SCALED RANGE OF ,F11.4,55H KILOFEET IS LESS THAN MINIMI
      1UM PERMISSIBLE VALUE, 0.115)
      END

```

23
 34
 45
 56
 67
 78
 89
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35

```

SUBROUTINE TIMHST(FTG,
1 NINTLO,NRFQ,NTMRSP,NTMGST,TIMEX,OMEGMX,INDSYM,RHOA,
1 VELG,TMGST,RHO,OMEGA,PLOTF,PLOTA,
2 AS,AA,AST,AAT,NACC,NG,NGL,PLOT,
1 PS,PA,PST,PAT,TIME)
C
C CMGN/XTF/NF(100)
EQUIVALENCE (NF(22),KPRBLS)
1 , (NF(28),IPLRL)
C
C (MMON/Z/Z/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
C
C DIMENSION VELG(1),TMGST(1),RHO(1)
C DIMENSION OMEGA(1),PLOTF(1),PLOTA(1)
C DIMENSION AS(1),AA(1),AST(1),AAT(1),PLOT(1)
C DIMENSION PS(2),NINTLO(1),PA(2),NINTLO(1)
C DIMENSION PST(NINTLO(1)),PAT(NINTLO(1))
C DIMENSION TIME(1)
C DIMENSION FTG(2,1)
C
C DATA TWOP/6.28318/
10 FORMAT(1H,20X,SYMMETRIC TIME RESPONSE - INTG LOADS /IHO,
1 20X,F8.4,RESPONSE SECS,2X,F8.4,HZ COUNTED)
11 FORMAT(1H,20X,ANTISYMMETRIC TIME RESPONSE - INTG LOADS /IHO,
1 20X,F8.4,RESPONSE SECS,2X,F8.4,HZ COUNTED)
20 FORMAT(1H,20X,SYMMETRIC FREQUENCY RESPONSE FUNCTION )
15 FORMAT(1H,20X,ANTISYMMETRIC FREQUENCY RESPONSE FUNCTION )
25 FORMAT(1H,20X,FREQ,4(2X,REAL-STA,13,2X,IMAG-STA,13))
40 FORMAT(1H,20X,TIME,4(2X,STA,13))
50 FORMAT(1H,F8.4,8(1X,E12.5))
C
INDSYM=1 IS A SYMMETRIC CASE
INDSYM=2 IS AN UNSYMMETRIC CASE
C
C IF (IPLRL.EQ.0) GO TO 100
CALL PLTI(NTMGST,TMGST,VELG,1)
WRITE(OUTPUT,5)
5 FORMAT(1H,20X,GUST TIME HISTORY AT AAS ORIGIN )
100 CONTINUE
C
CALL IPFFT(FTG,NTMGST,OMEGA,VELG,RHO,RHOA,NRFQ,PLOTF,PLOTA,
1 TMGST,CZERO,ALPHA,BETA)
C
NGL1 = IABS(NGL)
C
IF (NACC.EQ.0) GO TO 150
C
CALL RLOAD (5,NGL,AS,NOUT,NER)
IF (INDSYM.GT.1) CALL RLOAD (7,NGL,AA,NOUT,NER)
CALL ACCHST
1 ( FTG,NACC,NRFQ,NTMRSP,OMEGMX,INDSYM,OMEGA

```

```

TIMHST 2
TIMHST 3
TIMHST 4
TIMHST 5
TIMHST 6
TIMHST 7
TIMHST 8
TIMHST 9
TIMHST 10
TIMHST 11
TIMHST 12
TIMHST 13
TIMHST 14
TIMHST 15
TIMHST 16
TIMHST 17
TIMHST 18
TIMHST 19
TIMHST 20
TIMHST 21
TIMHST 22
TIMHST 23
TIMHST 24
TIMHST 25
TIMHST 26
TIMHST 27
TIMHST 28
TIMHST 29
TIMHST 30
TIMHST 31
TIMHST 32
TIMHST 33
TIMHST 34
TIMHST 35
TIMHST 36
TIMHST 37
TIMHST 38
TIMHST 39
TIMHST 40
TIMHST 41
TIMHST 42
TIMHST 43
TIMHST 44
TIMHST 45
TIMHST 46
TIMHST 47
TIMHST 48
TIMHST 49
TIMHST 50
TIMHST 51
TIMHST 52
TIMHST 53

```

```

2 , AS,AA,AST,AAT,TIME,TIMEMX,KPRHLS,NG,NGL,IPLBL,PLOT)
C 150 CONTINUE
CALL RDLOAD (B,NGLL,FS,NOUT,NFR)
IF (INDSYM.GT.1) CALL RDLOAD (9,NGLL,PA,NOUT,NFR)
C
DC 200 J=1,NFRFQ
AP=FTG(1,J)
AI=FTG(2,J)
NC 200 I=1,NINTLD
T=PS(1,I,J)*AR-PS(2,I,J)*AI
PS(2,I,J)=PS(1,I,J)*AI+PS(2,I,J)*AR
PS(1,I,J)=T
IF (INDSYM.EQ.1) GO TO 200
T=PA(1,I,J)*AR-PA(2,I,J)*AI
PA(2,I,J)=PA(1,I,J)*AI+PA(2,I,J)*AR
PA(1,I,J)=T
200 CONTINUE
C
IF (KPRBLS.EQ.0) GO TO 295
LINES=KRCW+1
JF=0
210 JS=JF+1
JF=JS+3
IF (JF.GT.NINTLD) JF=NINTLD
IF (LINES.LE.KRCW) GO TO 220
CALL HEADNG
WRITE (NOUT,20)
WRITE (NOUT,25) (J,J,J=JS,JF)
LINES=6
220 CONTINUE
NC 240 K=1,NFRFQ
LINES=LINES+1
IF (LINES.LE.KRCW) GO TO 230
CALL HEADNG
WRITE (NOUT,20)
WRITE (NOUT,25) (J,J,J=JS,JF)
LINES=6
230 CONTINUE
FRFQ=OMEGA(K1/TWCP1
240 WRITE (NOUT,50) FRFQ,(PS(1,J,K),PS(2,J,K),J=JS,JF)
IF (JF.GE.NINTLD) GO TO 250
J1=JF+1
J2=J1+3
WRITE (NOUT,25) (J,J,J=J1,J2)
LINES=LINES+2
GO TO 210
C 250 CONTINUE
IF (INDSYM.EQ.1) GO TO 295
C

```

TIMHST154
TIMHST155
TIMHST156
TIMHST157
TIMHST158
TIMHST159
TIMHST160
TIMHST161
TIMHST162
TIMHST163
TIMHST164
TIMHST165
TIMHST166
TIMHST167
TIMHST168
TIMHST169
TIMHST170
TIMHST171
TIMHST172
TIMHST173
TIMHST174
TIMHST175
TIMHST176
TIMHST177
TIMHST178
TIMHST179
TIMHST180
TIMHST181
TIMHST182
TIMHST183
TIMHST184
TIMHST185
TIMHST186
TIMHST187
TIMHST188
TIMHST189
TIMHST190
TIMHST191
TIMHST192
TIMHST193
TIMHST194
TIMHST195
TIMHST196
TIMHST197
TIMHST198
TIMHST199
TIMHST200
TIMHST201
TIMHST202
TIMHST203
TIMHST204
TIMHST205

TIMH SL06
 TIMH SL07
 TIMH SL08
 TIMH SL09
 TIMH SL10
 TIMH SL11
 TIMH SL12
 TIMH SL13
 TIMH SL14
 TIMH SL15
 TIMH SL16
 TIMH SL17
 TIMH SL18
 TIMH SL19
 TIMH SL20
 TIMH SL21
 TIMH SL22
 TIMH SL23
 TIMH SL24
 TIMH SL25
 TIMH SL26
 TIMH SL27
 TIMH SL28
 TIMH SL29
 TIMH SL30
 TIMH SL31
 TIMH SL32
 TIMH SL33
 TIMH SL34
 TIMH SL35
 TIMH SL36
 TIMH SL37
 TIMH SL38
 TIMH SL39
 TIMH SL40
 TIMH SL41
 TIMH SL42
 TIMH SL43
 TIMH SL44
 TIMH SL45
 TIMH SL46
 TIMH SL47
 TIMH SL48
 TIMH SL49
 TIMH SL50
 TIMH SL51
 TIMH SL52
 TIMH SL53
 TIMH SL54
 TIMH SL55
 TIMH SL56
 TIMH SL57

```

    LINES=KROW+1
    JF=0
    JS=JF+1
    JF=JS+3
    IF (JF.GT.NINTLD) JF=NINTLD
    IF (LINES.LE.KROW) GO TO 270
    CALL HEADNG
    WRITE (NOUT,15)
    WRITE (NOUT,25) (J,J,J=JS,JF)
    LINES=6
    CONTINUE
270 DO 290 K=1,NFREQ
    LINES=LINES+1
    IF (LINES.LE.KROW) GO TO 280
    CALL HEADNG
    WRITE (NOUT,15)
    WRITE (NOUT,25) (J,J,J=JS,JF)
    LINES=6
    CONTINUE
280 CONTINUE
    FREQ=OMEGA(K)/TWOPI
    WRITE (NOUT,50) FREQ, (PA(1,J,K),PA(2,J,K),J=JS,JF)
290 IF (JF.GE.NINTLD) GO TO 295
    J1=JF+1
    J2=J1+3
    WRITE (NOUT,25) (J,J,J=J1,J2)
    LINES=LINES+2
    GO TO 260
C 295 CONTINUE
C
C    CALL IFT(NFREQ,OMEGA,OMEGMX,0.,0,NINTLD,PS,
C    1 NTRSP,TIME,PST,TIMEMX)
C
C    IF (INDSYM.NE.1) CALL IFT(NFREQ,OMEGA,OMEGMX,0.,0,NINTLD,PA,
C    1 NTRSP,TIME,PAT,TIMEMX)
C
C    FRQMX=OMEGMX/TWOPI
C    IF (KPRBLS.EQ.0) GO TO 500
C    JF=0
C    LINES=KROW+1
C
C    JS=JF+1
C    JF=JS+7
C    IF (JF.GT.NINTLD) JF=NINTLD
C    IF (LINES.LE.KROW) GO TO 310
C    CALL HEADNG
C    WRITE (NOUT,10) TIMEMX,FRQMX
C    WRITE (NOUT,40) (J,J=JS,JF)
C    LINES=6
C    CONTINUE
310 DO 400 K=1,NTRSP
  
```

TIMH S158
 TIMH S159
 TIMH S160
 TIMH S161
 TIMH S162
 TIMH S163
 TIMH S164
 TIMH S165
 TIMH S166
 TIMH S167
 TIMH S168
 TIMH S169
 TIMH S170
 TIMH S171
 TIMH S172
 TIMH S173
 TIMH S174
 TIMH S175
 TIMH S176
 TIMH S177
 TIMH S178
 TIMH S179
 TIMH S180
 TIMH S181
 TIMH S182
 TIMH S183
 TIMH S184
 TIMH S185
 TIMH S186
 TIMH S187
 TIMH S188
 TIMH S189
 TIMH S190
 TIMH S191
 TIMH S192
 TIMH S193
 TIMH S194
 TIMH S195
 TIMH S196
 TIMH S197
 TIMH S198
 TIMH S199
 TIMH S200
 TIMH S201
 TIMH S202
 TIMH S203
 TIMH S204
 TIMH S205

```

    LINES=LINES+1
    IF (LINES.LE.KROW) GO TO 315
    CALL HEADNG
    WRITE (NCUT,40) (J,J=JS,JF)
    LINES=6
315 CONTINUE
400 WRITE (NCUT,50) TIME(K), (PST(J,K), J=JS,JF)
    IF (JF.GF.NINTLD) GO TO 410
    J1=JF+1
    J2=J1+7
    WRITE (NCUT,40) (J,J=J1,J2)
    LINES=LINES+2
    GO TO 300

C 410 CONTINUE
    IF (INDSYM.EQ.1) GO TO 500
    JF=0
    LINES=KFCW+1
    JS=JF+1
    JF=JS+7
    IF (JF.GT.NINTLD) JF=NINTLD
    IF (LINES.LE.KPCW) GO TO 320
    CALL HEADNG
    WRITE (NCUT,11) TIME(K), FRQMX
    WRITE (NCUT,40) (J,J=JS,JF)
    LINES=6
320 CONTINUE
    IF 430 K=1, NIMRSP
    LINES=LINES+1
    IF (LINES.LE.KRCW) GO TO 316
    CALL HEADNG
    WRITE (NCUT,40) (J,J=JS,JF)
    LINES=6
316 CONTINUE
430 WRITE (NCUT,50) TIME(K), (PAT(J,K), J=JS,JF)
    IF (JF.GF.NINTLD) GO TO 500
    J1=JF+1
    J2=J1+7
    WRITE (NCUT,40) (J,J=J1,J2)
    LINES=LINES+2
    GO TO 420

C 500 CONTINUE
C RETURN
END
  
```

```

SUBROUTINE TPEVAL(
1 TIME,RHCA,VSS,SLNTRG,SHA,SHB,SGA,SGOTP,<GRD,SRGOTP,CTP,
1 SCALR,SCALT,GUST,RHO)
2
3 THIS ROUTINE THANKS TO AL SHARP, AFWL AND VIHRA 4
4 SHA=SCALED HEIGHT ABOVE GRD OF REF POINT
5 SHB=SCALED HEIGHT ABOVE GRD OF BURST
6 SGA=SCALED GPD RANGE
7 SGOTP=SCALED GPD RANGE OF START PT OF TPP
8 SGOTP=SCALED GPD RANGE FROM BURST TO START PT OF TPP
9
10 COMMON/ZZ/HEADP(48),NIN,NOUT
11
12 DIMENSION FRPP(6),FRMR(75),SHBT(5),SRT(6),SRMT(15)
13
14 DATA FRPP/1.95,1.585,1.515,1.46,1.43,1.4028333/
15 DATA SRT/0.0,0.9,1.2,1.6,2.0,2.4/
16 DATA FRMR
17 DATA SRMT/0.6,0.8,1.0,1.1,1.2,1.4,1.6,2.0,2.2,2.4,3.4,4.4,6.4,50.0
18
19 DATA SHBT/0.0,0.4,0.7,1.0,1.4/
20 DATA FRMP/15#1.2,1.31,1.42,1.49,1.495,1.485,1.460,1.45,1.43
21
22 1 1.425,1.42,1.4,1.395,1.39,1.385,1.385,1.33,1.495,1.69,1.745
23 2 1.765,1.735,1.69,1.64,1.63,1.62,1.58,1.56,1.545,1.54,1.54
24 3 1.51,1.515,1.535,1.55,1.565,1.605,1.67,1.79,1.82,1.815,1.74
25 4 1.7,1.66,1.53,1.63,1.525,1.55,1.58,1.6,1.625,1.675,1.735
26 5 1.845,1.875,1.885,1.79,1.75,1.705,1.675,1.675/
27
28 K=1
29 RHCA=RHOA
30
31 IF (KGRD.EQ.1) GO TO 40
32
33 K=0
34
35 RTA=SCALR*SLNTRG
36 RK=1.0
37 IF (RTA.GT.0.55) RR=1.0+0.06*(RTA-0.55)**(1./3.)
38 RP=RTA*RP
39 RG=RTA
40 SCALT=SCALT
41 FACT=0.
42 CALL PRESS(RP,ZETA)
43
44 CONTINUE
45 T=0.
46 CALL TAR(RTA,T)
47 TMTA=TIME-SCALT*T
48 IF (TMTA.LT.-0.10F-04) GO TO 170

```

```

C
IF (TMTA.LT.0) TMTA=0.
CALL GUSRHC (RG,TMTA,ZETA,SCALTY,CUSTR,RHOR)
CUST=GUSTR*VSS
RHO=RHO+RHOR*RHQA
C
IF (KGRD*K.FQ.0) GO TO 140
GO TO 60
C
CONTINUE
SPIST=0.
T=TIME/SCALT
CALL TAR(SRIST,T)
IF (SPIST.LT.SHB) GO TO 10
C
IF (SHA.LT.0) GO TO 150
C
IF (SGA.LT.SGOTP) GO TO 50
C
SHTP=SGOTP*CTP*(SGA/SGOTP-1.)*1.6
IF (SHA.LT.SHTP) GO TO 130
C
CONTINUE
IF (SRIST.LT.SCALR*SLNTRG) GO TO 140
GO TO 20
C
CONTINUE
SPRS=SQRT(SGA**2+(SHA+SHB)**2)
IF (SRIST.GT.SFGOTP) GO TO 70
C
IF (SRIST.LT.SPRS) GO TO 140
GO TO 80
C
CONTINUE
CALL TPINT(SHB,SRIST,CTP,SGOTP,SHINT,SGINT)
IF (SGINT#2+(SHINT+SHB)**2.LT.SPRS**2) GO TO 140
C
CONTINUE
IF (SPRS.GT.SFGOTP) GO TO 120
C
IF (SPRS.GE.2.4) GO TO 90
C
CALL INTPPB(SPRS,0.,0.,6,0,0,SRT,0.,0.,FRRR,CRRRR)
GO TO 100
C
CONTINUE
CRRRR=1.147+0.614/SPRS
C
100 CONTINUE

```

TPE VAL54
TPE VAL55
TPE VAL56
TPE VAL57
TPE VAL58
TPE VAL59
TPE VAL60
TPE VAL61
TPE VAL62
TPE VAL63
TPE VAL64
TPE VAL65
TPE VAL66
TPE VAL67
TPE VAL68
TPE VAL69
TPE VAL70
TPE VAL71
TPE VAL72
TPE VAL73
TPE VAL74
TPE VAL75
TPE VAL76
TPE VAL77
TPE VAL78
TPE VAL79
TPE VAL80
TPE VAL81
TPE VAL82
TPE VAL83
TPE VAL84
TPE VAL85
TPE VAL86
TPE VAL87
TPE VAL88
TPE VAL89
TPE VAL90
TPE VAL91
TPE VAL92
TPE VAL93
TPE VAL94
TPE VAL95
TPE VAL96
TPE VAL97
TPE VAL98
TPE VAL99
TPE VAL100
TPE VAL101
TPE VAL102
TPE VAL103
TPE VAL104
TPE VAL105

TPE VAL06
TPE VAL07
TPE VAL08
TPE VAL09
TPE VAL10
TPE VAL11
TPE VAL12
TPE VAL13
TPE VAL14
TPE VAL15
TPE VAL16
TPE VAL17
TPE VAL18
TPE VAL19
TPE VAL20
TPE VAL21
TPE VAL22
TPE VAL23
TPE VAL24
TPE VAL25
TPE VAL26
TPE VAL27
TPE VAL28
TPE VAL29
TPE VAL30
TPE VAL31
TPE VAL32
TPE VAL33
TPE VAL34
TPE VAL35
TPE VAL36
TPE VAL37
TPE VAL38
TPE VAL39
TPE VAL40
TPE VAL41
TPE VAL42
TPE VAL43
TPE VAL44
TPE VAL45
TPE VAL46
TPE VAL47
TPE VAL48
TPE VAL49
TPE VAL50
TPE VAL51
TPE VAL52
TPE VAL53
TPE VAL54
TPE VAL55
TPE VAL56
TPE VAL57

```

R=SRMS/CRFRMR
CALL PRESS(R,ZETAT)
RP=SRMS
RTA=RP
RG=RP

C 110 CONTINUE
SCALTT=SCALT
RR=1.0
IF (RTA.GT.0.55)RR=1.0+0.06*(RTA-0.55)**(1./3.)
RP=RP*RR
CALL PRESS(RP,ZETAT)
ZETA=ZETAT-ZETAT
FACT=2.0
K=0
GO TO 30

C 120 CONTINUE
SH=-SHB
CALL TPINT(SH,SRPS,CTP,SGOTP,SHINT,SGINT)
SPMS=SQRT(SGINT**2+(SHINT-SHB)**2)
SRMS=SQRT(SGINT**2+SHINT**2)
SH=AMINI(SHB,1.4)
CALL INTRPB(SH,SRMS,0.,5,15,0,SHBT,SRMT,0.,FRMR,CRFRMR)
P=SRMS/CRFRMR
CALL PRESS(R,ZETATM)
ZETAT=AMINI(ZETATP,ZETATM)
RTA=SRIS
RP=SRIS
RG=SRIS
GO TO 110

C 130 CONTINUE
CALL TPINT(SHB,SRIS,CTP,SGOTP,SHINT,SGINT)
SPMS=SQRT(SGA**2+SHA**2)
IF (SGINT**2+SHINT**2.LT.SRMS**2)GO TO 140
SH=0.
CALL TPINT(SH,SRMS,CTP,SGOTP,SHINT,SGINT)
SRIS=SQRT(SGINT**2+(SHINT-SHB)**2)
SH=AMINI(SHB,1.4)
CALL INTRPB(SH,SRMS,0.,5,15,0,SHBT,SRMT,0.,FRMR,CRFRMR)
K=SRMS/CRFRMR
CALL PRESS(R,ZETAT)
ZETA=AMINI(ZETATP,ZETA)
RTA=SRIS
RG=SPMS/1.2
SCALTT=1.2*SCALT
FACT=1.
K=0
GO TO 30

```

TPE VAL 58
TPE VAL 59
TPE VAL 60
TPE VAL 61
TPE VAL 62
TPE VAL 63
TPE VAL 64
TPE VAL 65
TPE VAL 66
TPE VAL 67
TPE VAL 68
TPE VAL 69
TPE VAL 70
TPE VAL 71
TPE VAL 72
TPE VAL 73

```

140 CONTINUE
    RETURN

C
150 CONTINUE
    WRITE (INOUT,160)
160 FORMAT(1H0,20X, AAS ORIGIN BELOW GRD LEVEL - CHECK FLIGHT PLAN )
    NCR=1
    RETURN

C
170 CONTINUE
    WRITE (INOUT,180) TMTA
180 FORMAT(1H0,20X, A GOOF, DELTA T AFTER INTERCEPT IS NEG.= ,E16.8)
    TIME=TIME+0.0010
    GO TO 30

C
    END

```

TPINT 2
TPINT 3
TPINT 4
TPINT 5
TPINT 6
TPINT 7
TPINT 8
TPINT 9
TPINT 10
TPINT 11
TPINT 12
TPINT 13
TPINT 14
TPINT 15
TPINT 16
TPINT 17
TPINT 18
TPINT 19
TPINT 20
TPINT 21
TPINT 22
TPINT 23
TPINT 24

SUBROUTINE TPINT (H,R,C,GO,HI,GI)
GIVEN H, R, C, GO, SUBROUTINE TPINT RETURNS HI AND GI.
HI IS THE SCALED HEIGHT OF THE CENTER OF THE CIRCLE
WHOSE INTERSECTION WITH THE TRIPLE POINT PATH IS SOUGHT.
R IS THE RADIUS OF THE CIRCLE.
C IS THE CONSTANT IN THE EQUATION FOR THE HEIGHT OF THE TRIPLE
POINT PATH, HT, AS A FUNCTION OF GROUND RANGE, G.
GO IS THE GROUND RANGE AT WHICH THE TRIPLE POINT PATH STARTS.
GI IS THE HEIGHT OF THE INTERSECTION.
HI IS THE GROUND RANGE OF THE INTERSECTION.
D=C
T=D*(D*GO+H)
G=(T+SQRT(T**2-(1.0+D**2)*((D*GO+H)**2-R**2)))/(1.0+D**2)
HT=C*GO*(G/GO-1.0)**1.6
RT=SQRT(G**2+(HT-H)**2)
IF (ABS(RT/R-1.0).LE.0.000001) GO TO 20
D=H*Y/(G-GI)
GI TO 10
HI=HT
GI=G
RETURN
END

C 10
C 20

```

SUBROUTINE TRFFT(FTG,
1 NT,OMEGA,VELG,PHO,RHOA,NFREQ,PLOTF,PLOTA,TIME,GZERO,ALPHA,BETA)
C
COMMON/ZZZ/CASE(49),NIN,NOUT,KROW,LINES,IPRNT,NER
C
COMMON/XTF/NF(100)
EQUIVALENCE (NF(22),IPREQ)
1 , (NF(1),AMACH), (NF(2),GAMX)
1 , (NF(91),FINFRQ)
2 , (NF(89),TAU)
C
C
DIMENSION OMEGA(1),VELG(1),RHO(1)
DIMENSION PLOTF(1)
DIMENSION PLOTA(1)
DIMENSION TIME(1)
DIMENSION FTG(2,1)
C
DATA TWOPI/6.28318/
DATA FPV/0.053/
C
1 FORMAT(1H0,20X, TRANSFORMED FREQUENCY SPECTRUM /1H0,
5 20X,I4, FREQUENCIES ,I4, TIME POINTS )
1 NFRQ ,2X, FREQ , REAL , IMAG ,
1 NFRQ ,2X, MOD ,2X, REAL , IMAG ,
1 NFRQ ,2X, MOD ,
2 FORMAT(1H ,I4,2X,F8.3,3E14.5,2X,I4,2X,F8.3,3E14.5)
C
NFREQ=NFRQ/2
C
II=0
ON 20 I=1,NT
VELG(1)=VELG(1)*RHO(1)/RHOA
IF (II.GT.0) GO TO 20
IF (VELG(1).LT.0) II=1
CONTINUE
20
C
ITEST = 1
IF (II.EQ.0) GO TO 30
C
GZERO=VELG(1)
T1 = TIME(II-1) + TAU
T2 = TIME(II) + TAU
V1=VELG(II-1)
V2=VELG(II)
T0 = TIME(II-1) + TAU + (T2-T1)*(V1-V2)
V0=0.0
T1 = TIME(II/2) + TAU
V1=VELG(II/2)
C

```

```

TRFFT 2
TRFFT 3
TRFFT 4
TRFFT 5
TRFFT 6
TRFFT 7
TRFFT 8
TRFFT 9
TRFFT 10
TRFFT 11
TRFFT 12
TRFFT 13
TRFFT 14
TRFFT 15
TRFFT 16
TRFFT 17
TRFFT 18
TRFFT 19
TRFFT 20
TRFFT 21
TRFFT 22
TRFFT 23
TRFFT 24
TRFFT 25
TRFFT 26
TRFFT 27
TRFFT 28
TRFFT 29
TRFFT 30
TRFFT 31
TRFFT 32
TRFFT 33
TRFFT 34
TRFFT 35
TRFFT 36
TRFFT 37
TRFFT 38
TRFFT 39
TRFFT 40
TRFFT 41
TRFFT 42
TRFFT 43
TRFFT 44
TRFFT 45
TRFFT 46
TRFFT 47
TRFFT 48
TRFFT 49
TRFFT 50
TRFFT 51
TRFFT 52
TRFFT 53

```

```

TRFF T 54
TRFF T 55
TRFF T 56
TRFF T 57
TRFF T 58
TRFF T 59
TRFF T 60
TRFF T 61
TRFF T 62
TRFF T 63
TRFF T 64
TRFF T 65
TRFF T 66
TRFF T 67
TRFF T 68
TRFF T 69
TRFF T 70
TRFF T 71
TRFF T 72
TRFF T 73
TRFF T 74
TRFF T 75
TRFF T 76
TRFF T 77
TRFF T 78
TRFF T 79
TRFF T 80
TRFF T 81
TRFF T 82
TRFF T 83
TRFF T 84
TRFF T 85
TRFF T 86
TRFF T 87
TRFF T 88
TRFF T 89
TRFF T 90
TRFF T 91
TRFF T 92
TRFF T 93
TRFF T 94
TRFF T 95
TRFF T 96
TRFF T 97
TRFF T 98
TRFF T 99
TRFF T 100
TRFF T 101
TRFF T 102
TRFF T 103
TRFF T 104
TRFF T 105

```

```

C
GAMMA=ALOG(2.0)/TO
ALPHA=-((ALOG(VI/GZERO)-ALOG(2.0-2.0**((11/TO)))/T1
RFTA=ALPHA-GAMMA
C
T1 = T1 - TAU
T0 = T0 - TAU
NCW CHECK FIT
ITEST=0
C
DO 25 I=2,NT,2
VTEST=GZERO*(2.0*FXP(-ALPHA*TIME(I))-EXP(-BETA*TIME(I)))
IF(ABS(VTEST-VELG(I))/VTEST).GT.EPV) ITEST=1
CONTINUE
25
C
IF(ITEST.FQ.0)GO TO 36
C
CONTINUE
DO 35 J=1,NREQ
FTG(1,J)=0.
FTG(2,J)=0.
CMEGI=1.0/OMEGA(J)
ANG1=OMEGA(J)*TIME(1)
C1=COS(ANG1)
S1=SIN(ANG1)
DO 34 I=2,NT
DEL=1.0/(TIME(I)-TIME(I-1))
R=(VELG(I)-VELG(I-1))*DEL
ANG2=OMEGA(J)*TIME(I)
S2=SIN(ANG2)
C2=COS(ANG2)
RP=OMEGI*(VELG(I)*S2-VELG(I-1)*S1)+B*OMEGI*OMEGI*(C2-C1)
R1=-OMEGI*(VELG(I)*C2-VELG(I-1)*C1)+B*OMEGI*OMEGI*(S2-S1)
FTG(1,J)=FTG(1,J)+RR
FTG(2,J)=FTG(2,J)-R1
C1=C2
S1=S2
CONTINUE
35
CONTINUE
ALPHA=0.0
RFTA=0.0
C
36
CONTINUE
C
IF(ITEST.NE.0)GO TO 45
DO 40 J=1,NREQ
C = COS(OMEGA(J)*TAU)
S = SIN(OMEGA(J)*TAU)
DENR=ALPHA**2+CMEGA(J)**2
DENR=BETA**2+CMEGA(J)**2
F1 = (2.*ALPHA/DENR - BETA/DENR)*GZERO
F2 = (2./DENR-1.0/DENR)*GZERO*OMEGA(J)
FTG(1,J) = F1*C + F2*S
C
C

```

```

TRFF T106
TRFF T107
TRFF T108
TRFF T109
TRFF T110
TRFF T111
TRFF T112
TRFF T113
TRFF T114
TRFF T115
TRFF T116
TRFF T117
TRFF T118
TRFF T119
TRFF T120
TRFF T121
TRFF T122
TRFF T123
TRFF T124
TRFF T125
TRFF T126
TRFF T127
TRFF T128
TRFF T129
TRFF T130
TRFF T131
TRFF T132
TRFF T133
TRFF T134
TRFF T135
TRFF T136
TRFF T137
TRFF T138
TRFF T139
TRFF T140
TRFF T141
TRFF T142
TRFF T143
TRFF T144
TRFF T145
TRFF T146
TRFF T147
TRFF T148
TRFF T149
TRFF T150
TRFF T151
TRFF T152

```

```

40 FTG(2,J) = F1*S - F2*C
C CONTINUE
45 CONTINUE
C
A2=SQRT(FTG(1,NFREQ)**2+FTG(2,NFREQ)**2)
A1=SQRT(FTG(1,NFREQ-5)**2+FTG(2,NFREQ-5)**2)
FINFRQ=-(A1*OMEGA(NFREQ)-A2*OMEGA(NFREQ-5))/(A2-A1)
IF (FINFRQ.LT.0.0) FINFRQ = OMEGA(NFREQ) + 100.
C
IF (IPREQ.EQ.0) GO TO 50
CALL HEADNG
WRITE (NOUT,1) NFREQ,NT
*WRITE (NOUT,4) TIME(1),GZFRQ,T1,V1,T0,V0,TIME(NT),VFLG(NT),
1 ALPHA,BETA
4 FCRMAT(1H0,24X,TIME,3X,V*RH0/1H,20X,F8.4,1X,F8.4/
1 1H,20X,F8.4,1X,F8.4/
1 1H,20X,F8.4,1X,F8.4/
1 1H0,20X,ALPHA=.E13.6,BETA=.E13.6)
WRITE (NOUT,5)
LINES=13
CC 60 J=1,NFREQ
JJ=J+NFREQP
PLOT( J )=OMEGA( J )/TWOPI
PLOT( JJ)=OMEGA( JJ )/TWOPI
PLOT( J )=SQRT(FTG(1,J)**2+FTG(2,J)**2)
PLOT( JJ)=SQRT(FTG(1,JJ)**2+FTG(2,JJ)**2)
IF (LINES.LT.KRCW) GO TO 55
CALL HEADNG
WRITE (NOUT,1) NFREQ,NT
WRITE (NOUT,5)
LINES=7
LINES=LINES+1
*WRITE (NOUT,2) J,PLOT( J ),FTG(1,J),FTG(2,J),PLOT( J ),
1 JJ,PLOT( JJ ),FTG(1,JJ),FTG(2,JJ),PLOT( JJ )
C CONTINUE
55 CONTINUE
FINFRQ=FINFRQ/TWOPI
C
WRITE (NOUT,3) FINFRQ
FORMAT(1H,20X,EST.FREQ.FOR 0 MOD. OF GUST FUNCTION = ,
1 F10.2,4Z)
C
PRTURN
END

```

[illegible]

```

BLINT 54
BLINT 55
BLINT 56
BLINT 57
BLINT 58
BLINT 59
BLINT 60
BLINT 61
BLINT 62
BLINT 63
BLINT 64
BLINT 65
BLINT 66
BLINT 67
BLINT 68
BLINT 69
BLINT 70
BLINT 71
BLINT 72

```

```

ZINT=(GFCMRD(13)+GFCMRD(16))/2.0
YINTL=-YINT
ALR=GAMX*XINT+GAMY*YINT+GAMZ*ZINT
IF(ALR.GT.AL1)GO TO 35
ALI=ALR
X=XINT
Y=YINT
Z=ZINT
CONTINUE
ALL=GAMX*XINT+GAMY*YINTL+GAMZ*ZINT
IF(ALL.GT.AL1)GO TO 40
ALI=ALL
X=XINT
Y=YINTL
Z=ZINT
CONTINUE
RETURN
END

```

35

40

C

```

C
SUBROUTINE CRIGID
COMMON NAA,A(1)
COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINES,IPRNT,NER
COMMON /ATF/NF(100)
EQUIVALENCE (NF(6),NSYM), (NF(9),NFRFQ), (NF(22),IPREQ)
1 , (NF(42),NG)
2 , (NF(44),MSBE)
3 , (NF(43),MBOX), (NF(97),DELT)
      , (NF(62),TIMEMX),
10 FORMAT (6I12)
20 FORMAT (6F12.0)
30 FORMAT (1H1,20X,*RUN DATA FOR RIGID*
1 / 1H0,20X,*NUMBER OF TIME POINTS
2 / 1H ,20X,*NUMBER OF AERU BOXES
3 / 1H ,20X,*NUMBER OF SLFNDER BODY ELEMENTS
5 / 1H ,20X,*GUST ORIENTATION
6 / 1H ,20X,*PLOT FLAG
7 / 1H ,20X,*CHECK PRINT FLAG
8 / 1H ,20X,*MAX TIME
9 / 1H ,20X,*DELTA TIME
40 FORMAT (1H0,20X,*FREQUENCIES FOR RESPONSE SOLUTION*
1 / 1H0,*ITM*,6X,*LWR F*,6X,*UPR F*,6X,*DEL F*)
41 FORMAT (1H ,13,3F10.4)
C
READ (NIN,10) NTMGST, NBOX, NSBE, NCR, IPLOT, IPREQ
READ (NIN,20) TIMEMX, DELT
WRITE (NOUT,30) NTMGST,NBOX,NSBE,NCR,IPLOT,IPREQ,TIMEMX,DELT
NFORCE = NBOX + NSBE
C
READ IN FREQUENCIES FOR RESPONSE SOLUTION
C
READ (NIN,10) NFRGR
I = 0
WRITE (NOUT,40)
DO 120 N=1,NFRGR
READ (NIN,20) F1,F2,DF
WRITE (NOUT,41) N,F1,F2,DF
I = I+1
A(I) = F1
110 I = I+1
A(I) = A(I-1) + DF
IF (A(I).LT.F2) GO TO 110
IF (A(I).GT.F2) I=I-1
120 NFRGR = I
C
NTMRSP=0
NT=TIMEMX/DELT+1
DELT=DELT
IND=0
TI=0.

```

CRIGID 2
 CRIGID 3
 CRIGID 4
 ZZZ 2
 CRIGID 6
 CRIGID 7
 CRIGID 8
 CRIGID 9
 CRIGID 10
 CRIGID 11
 CRIGID 12
 CRIGID 13
 CRIGID 14
 CRIGID 15
 CRIGID 16
 CRIGID 17
 CRIGID 18
 CRIGID 19
 CRIGID 20
 CRIGID 21
 CRIGID 22
 CRIGID 23
 CRIGID 24
 CRIGID 25
 CRIGID 26
 CRIGID 27
 CRIGID 28
 CRIGID 29
 CRIGID 30
 CRIGID 31
 CRIGID 32
 CRIGID 33
 CRIGID 34
 CRIGID 35
 CRIGID 36
 CRIGID 37
 CRIGID 38
 CRIGID 39
 CRIGID 40
 CRIGID 41
 CRIGID 42
 CRIGID 43
 CRIGID 44
 CRIGID 45
 CRIGID 46
 CRIGID 47
 CRIGID 48
 CRIGID 49
 CRIGID 50
 CRIGID 51
 CRIGID 52
 CRIGID 53

CRIGID54
CRIGID55
CRIGID56
CRIGID57
CRIGID58
CRIGID59
CRIGID60
CRIGID61
CRIGID62
CRIGID63
CRIGID64
CRIGID65
CRIGID66
CRIGID67
CRIGID68
CRIGID69
CRIGID70
CRIGID71
CRIGID72
CRIGID73
CRIGID74
CRIGID75
CRIGID76
CRIGID77
CRIGID78
CRIGID79
CRIGID80
CRIGID81
CRIGID82
CRIGID83
CRIGID84
CRIGID85
CRIGID86
CRIGID87
CRIGID88
CRIGID89
CRIGID90
CRIGID91
CRIGID92
CRIGID93
CRIGID94
CRIGID95
CRIGID96
CRIGID97
CRIGID98
CRIGID99
CRIGI100
CRIGI101
CRIGI102
CRIGI103
CRIGI104
CRIGI105

```

00 200 I=1,NT
   TI=TI+DELT
   IF(TI.GT.TIMEMX)GO TO 250
   NTMRSP=NTMRSP+1
   IF(IND.GT.1)GO TO 200
   IF(TI.LT.0.25)GO TO 200
   IF(TI.LE.1.0)GO TO 150
   IND=2
   TI=TI-DELT
   DELTT=DELT*2.0
   TI=TI+DELT
   GO TO 200
150 IF(IND.EQ.1)GO TO 200
   IND=1
   TI=TI-DELT
   DELTT=DELT*5.0
   TI=TI+DELT
200 CONTINUE
250 CONTINUE
   NTMRSP=NTMRSP+1
   IF(TI-DELT.LT.TIMEMX.AND.ABS(TI-DELT-TIMEMX).GT.0.010)
1 NTMRSP=NTMRSP+1

```

C

```

L1 = 1
L2 = L1
L3 = L2
L4 = L3
L5 = L4
L6 = L5
L7 = L6
L8 = L7
L20 = L8
L21 = L20
L22 = L21
L23 = L22
L9 = L23
L10 = L9
L10A = L10
L11 = L10
L12 = L11
L13 = L12
L14 = L13
L15 = L14
L16 = L15
L17 = L16
L18 = L17
L19 = L18
   NFREQ
   + MAXO(NTMGST,NTMRSP)
   + NFREQ
   + NFREQ
   + 2*NFREQ
   + 2*NFREQ*NFREQ
   + NBOX
   + *NSRE
   + NSYM
   + 2*NFREQ*NK
   + NK
   + NFREQ*NK
   + MBOX*5
   + MSRE*6
   + NTMGST
   + NTMGST
   + NTMGST
   + NFREQ
   + MAXO(NFREQ,NTMRSP)

```

C

```

CALL RIGID (NTMGST,NBOX,NFREQ,NFORCE,NOR,IPLOT,NTMRSP
1 , A(L1),A(L2),A(L3),A(L4),A(L5)
2 , A(L6),A(L7),A(L8),A(L9),A(L10),A(L11),A(L12),MBOX,MSBE
3 , A(L13),A(L14),A(L15),A(L16))

```

4 : A(L17),A(L18),A(L19)
5 : A(L20),A(L21),A(L22),A(L23))
C
RETURN
END

CRIGI106
CRIGI107
CRIGI108
CRIGI109
CRIGI110

```

C      SUBROUTINE GSINT (NFKEQ,NFORCE,NK,IPREQ
1,OMEGA,CP,FI,VOBWS,RS,TANS)
C
C      DIMENSION OMEGA(1)
C      COMPLEX CP(NFORCE,1), FI(NFORCE,1)
C      DIMENSION VOBWS(1), RS(1), TANS(1)
C
C      COMMON /ATF/NF(100)
C      EQUIVALENCE (NF(12),BZFRU), (NF(18),VIFPS), (NF(100),SIZFCT)
C
C      NKI = MINO(NK,10)
C      ISETA = 1
C      IF (VOBWS(NK).GT.VOBWS(1)) ISETA = -1
C      IMO = 0
C      OMEGF = VIFPS*12/(BZERO*SIZFCT)
C
C      DO 500 K=1,NFKEQ
C      VOBW = OMEGF/OMEGA(K)
C
C      FIND OUT IF WE NEED NEW COEFS
C      IREGG = 0
C      TESTV = 1.E6
C      DO 10 I=1,NK
C      TEST = ABS(VOBWS(I)-VOBW)
C      IF (TEST.GT.TESTV) GO TO 10
C      TESTV = TEST
C      IM = I
C10 CONTINUE
C
C      IL = IM-ISETA
C      IF (NKI.GT.4) IL=IL-ISETA
C      IH = IL+(NKI-1)*ISETA
C
C20 IF (IL.GE.1.AND.ISETA.GT.0) GO TO 30
C      IF (IL.LF.NK.AND.ISETA.LT.0) GO TO 30
C      IH = IH+ISETA
C      IM = IM+ISETA
C      IL = IL+ISETA
C      GO TO 20
C
C30 IF (IH.LF.NK.AND.ISETA.GT.0) GO TO 40
C      IF (IH.GE.1.AND.ISETA.LT.0) GO TO 40
C      IH = IH-ISETA
C      IM = IM-ISETA
C      IL = IL-ISETA
C      GO TO 20
C
C40 IF (IM.NF.IMO) IREGG = 1
C      IMO = IM
C
C      CALL INTRP2 (IZRO,VOBW,CP(1,K)
1,VOBWS(IL),FI(1,IL),NFORCE,NKI,RS,TANS,IREGG,IPREQ,K)

```

GSINT 2
 GSINT 3
 GSINT 4
 GSINT 5
 GSINT 6
 GSINT 7
 GSINT 8
 GSINT 9
 GSINT 10
 GSINT 11
 GSINT 12
 GSINT 13
 GSINT 14
 GSINT 15
 GSINT 16
 GSINT 17
 GSINT 18
 GSINT 19
 GSINT 20
 GSINT 21
 GSINT 22
 GSINT 23
 GSINT 24
 GSINT 25
 GSINT 26
 GSINT 27
 GSINT 28
 GSINT 29
 GSINT 30
 GSINT 31
 GSINT 32
 GSINT 33
 GSINT 34
 GSINT 35
 GSINT 36
 GSINT 37
 GSINT 38
 GSINT 39
 GSINT 40
 GSINT 41
 GSINT 42
 GSINT 43
 GSINT 44
 GSINT 45
 GSINT 46
 GSINT 47
 GSINT 48
 GSINT 49
 GSINT 50
 GSINT 51
 GSINT 52
 GSINT 53

GSINT 54
GSINT 55
GSINT 56
GSINT 57

C 500 CONTINUE
RETURN
END

```

C      SUBROUTINE INTRP2 (IZRU,VOBW,FGUST
1      , VOBWI,FI,NM,NK,RS,TANS,IREQ,IPREQ,J)
C
C      DIMENSION VOBWI(1)
C      DIMENSION FI(2,NM,1)
C      DIMENSION RS(NM,1),TANS(NM,1)
C      DIMENSION FGUST(2,1)
C      DIMENSION COEF(10), RK(10), C(13,10)
C
C      COMMON /ZZZ/CASE(48),NIN,NOUT,KROW,LINES,IPRNT,NER
C
C      DATA EPRV/0.001/
C      DATA EPC /0.1E-5/
C
C      DATA EP/1.0E-02/
C
C      NM = TOTAL AERO ELEMENTS = NBOX+2*NSBETO
C
10  FORMAT(1H0,20X, GUST INTERPOLATION ELEMENTS /1H0,
1  ELEM NG NK ,12X, RK ,12X, RI ,11X, MOD ,11X, TAN )
20  FORMAT(1H ,314,4E14.6)
C
DO 30 I=1,NK
K = I
IF (ABS(1.-VOBWI(1)/VOBW).LT.EPRV) GO TO 100
COEF(I) = 0.0
30  RK(I) = 1.0/VOBWI(1)
IF (IRECG.EQ.1) CALL FORMC (RK,NK,C,IPREQ)
RKIN = 1.0/VOBW
CALL COEFF (RKIN,RK,NK,C,COEF)
DO 35 L=1,NK
35  IF (ABS(COEF(L)).LT.EPC) COEF(L)=0.0
39  CONTINUE
C
C      K=0
C
DO 40 I=1,NK
IF(COEF(I).EQ.1.0)K=I
40  CONTINUE
C
IF(K.NE.0)GO TO 100
C
C
C
IF (IRECG.EQ.0) GO TO 120
ICNT=0
SET UP COUNT FOR ZERO GUST LOADS
IF(IPREQ.EQ.0)GO TO 47
C

```

```

INTRP2 2
INTRP2 3
INTRP2 4
INTRP2 5
INTRP2 6
INTRP2 7
INTRP2 8
INTRP2 9
INTRP2 10
INTRP2 11
INTRP2 12
INTRP2 13
INTRP2 14
INTRP2 15
INTRP2 16
INTRP2 17
INTRP2 18
INTRP2 19
INTRP2 20
INTRP2 21
INTRP2 22
INTRP2 23
INTRP2 24
INTRP2 25
INTRP2 26
INTRP2 27
INTRP2 28
INTRP2 29
INTRP2 30
INTRP2 31
INTRP2 32
INTRP2 33
INTRP2 34
INTRP2 35
INTRP2 36
INTRP2 37
INTRP2 38
INTRP2 39
INTRP2 40
INTRP2 41
INTRP2 42
INTRP2 43
INTRP2 44
INTRP2 45
INTRP2 46
INTRP2 47
INTRP2 48
INTRP2 49
INTRP2 50
INTRP2 51
INTRP2 52
INTRP2 53

```

```

C 47 CALL HEADNG
      WRITE(NOUT,10)
C     CONTINUE
      DO 90 I=1,NM
      DO 80 K=1,NK
      RR=FI(1,I,K)
      RI=FI(2,I,K)
      IF (ABS(RR).LT.EP)RR=0.
      IF (ABS(RI).LT.EP)RI=0.
      RT=SQRT(KR**2+RI**2)
      RS(I,K)=RR
      TANS(I,K)=RI
      IF (IPREQ.NE.0)WRITE(NOUT,20)I,J,K,RR,KI,KI,KT
      IF (RR.NE.0)GO TO 80
      IF (RI.NE.0)GO TO 80
      IF (K.GE.2)GO TO 85
      GO TO 80
C 80 CONTINUE
C
C 85 GO TO 90
      CONTINUE
      DO 86 K=1,NK
      TANS(I,K)=0.
      RS(I,K)=0.
      ICNT=ICNT+NK
C 86
C 90 CONTINUE
      IREG=1
C
C     IF (ICNT.GF.NK*NM)IZRO=1
C
C     IF (IZRO.EQ.1)GO TO 150
      GO TO 120
C 100 CONTINUE
      DO 110 I=1,NM
      FREAL=FI(1,I,K)
      FIMAG=FI(2,I,K)
      FGUST(1,I)=FREAL
      FGUST(2,I)=FIMAG
C 110 CONTINUE
C
C     GO TO 150
C
C 120 CONTINUE
      DO 140 I=1,NM

```

```

INTRP254
INTRP255
INTRP256
INTRP257
INTRP258
INTRP259
INTRP260
INTRP261
INTRP262
INTRP263
INTRP264
INTRP265
INTRP266
INTRP267
INTRP268
INTRP269
INTRP270
INTRP271
INTRP272
INTRP273
INTRP274
INTRP275
INTRP276
INTRP277
INTRP278
INTRP279
INTRP280
INTRP281
INTRP282
INTRP283
INTRP284
INTRP285
INTRP286
INTRP287
INTRP288
INTRP289
INTRP290
INTRP291
INTRP292
INTRP293
INTRP294
INTRP295
INTRP296
INTRP297
INTRP298
INTRP299
INTRP100
INTRP101
INTRP102
INTRP103
INTRP104
INTRP105

```

INTRP106
INTRP107
INTRP108
INTRP109
INTRP110
INTRP111
INTRP112
INTRP113
INTRP114
INTRP115
INTRP116
INTRP117
INTRP118
INTRP119
INTRP120
INTRP121
INTRP122
INTRP123
INTRP124
INTRP125
INTRP126
INTRP127
INTRP128
INTRP129
INTRP130
INTRP131
INTRP132
INTRP133

```

R=0
THETA=0
DO 130 K=1,NK
  R=R+COEF(K)*RS(I,K)
  THETA=THETA+COEF(K)*TANS(I,K)
C
  FREAL=R
  FIMAG=THETA
  FGUST(1,I)=FREAL
  FGUST(2,I)=FIMAG
C
  140 CONTINUE
C
  150 CONTINUE
C
  IF(IPREQ.EQ.0)GO TO 160
  WRITE (NOUT,36) (RK(K),K=1,NK)
  WRITE (NOUT,37) (COEF(K),K=1,NK)
  36 FORMAT (1H0,*INTERPOLATION COEFFICIENTS FOR K=*,F16.5)
  37 FORMAT (1H,*K=*,10F10.4)
  38 FORMAT (1H,*COEF=*,10F10.4)
C
  160 CONTINUE
C
  RETURN
C
  END

```

```

SUBROUTINE RIGID (NIMGSI,NBOX,NSHE,NFORCE,NOP,IPLLOT,NIMRSP
1 , OMEGA, TIME, GEOM, CPT, CP
2 , IROXES, ISRES, Q, GEOP, DB, MBCX, MSBE
3 , DP, KHU, VG, PLOTF, PLOTA, FTC, P
4 , FI, VUBWS, RS, TANS )

```

```

C
DIMENSION OMEGA(1), TIME(1), GEOM(1)
COMPLEX CPT(1), CP(NFORCE,1)
DIMENSION IROXES(1), ISRES(2,1), Q(1), GEOP(MBOX,1), GEOS(MSBE,1)
COMPLEX D(MBOX,1), DB(MSBE,1)
DIMENSION CR(3,1)
DIMENSION DP(1), RHO(1), VG(1), PLOTF(1), PLOTA(1)
COMPLEX FTC(1)
DIMENSION P(NFORCE,1)
COMPLEX FI(NFORCE,1)
DIMENSION VUBWS(1), RS(1), TANS(1)

```

```

COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINES,IPRNT,NER

```

```

C
COMMON /XTF/NF(100)
EQUIVALENCE (NF(1),AMACH), (NF(3),VSS)
EQUIVALENCE (NF(6),NSYM), (NF(9),NFREQ), (NF(22),IPREQ)
, (NF(12),BZERO), (NF(13),RHO0), (NF(14),SIGMA)
1 , (NF(18),VTFPS)
2 , (NF(41),NK)
3 , (NF(42),NG)
4 , (NF(62),TIME MX), (NF(97),DELT)
9 , (NF(100),SIZECT)

```

```

DATA TWOP1/6.2831853/

```

```

C
C
3 FORMAT (1H0,20X,*RIGID TIME RESPONSE PARAMETERS*

```

```

1 /1H,20X,*ORIENTATION NO. =*,I4
2 /1H,20X,*MAX. RESP. TIME =*,F10.4
2 /1H,20X,*NO. TIME RSP. PTS=*,I5
2 /1H,20X,*GAMX =*,E12.5
2 /1H,20X,*GAMY =*,E12.5
2 /1H,20X,*GAMZ =*,E12.5
2 /1H,20X,*TAU =*,F10.4,* SFCS*
2 /1H,20X,*XINT =*,F10.2
2 /1H,20X,*YINT =*,F10.2
2 /1H,20X,*ZINT =*,F10.2

```

```

10 FORMAT (6I12)
20 FORMAT (6F12.0)
30 FORMAT (1H0,3X1H,9X7HTIME(1),11X5HDP(1),10X6HRHO(1),11X5HWG(1)
1 // (1X,I4,E16.6) )
31 FORMAT (1H0,7X3HSTA,7X1HX,9X1HY,9X1HZ,6X4HAREA)
32 FORMAT (1X,2I10,10X4F10.2)
33 FORMAT (1H0,7X3HSTA,6X4HBDY,1X9HDIRECTION,7X1HX,9X1HY,9X1HZ
1 , 6X6HLENGTH)
34 FORMAT (1X,3I10,4F10.2)
38 FORMAT (1H0,*INPUT GENERALIZED COORDINATES*
1 // (1X,I4,E16.6) )

```

RIGID 2
RIGID 3
RIGID 4
RIGID 5
RIGID 6
RIGID 7
RIGID 8
RIGID 9
RIGID 10
RIGID 11
RIGID 12
RIGID 13
RIGID 14
RIGID 15
RIGID 16
RIGID 17
RIGID 18
RIGID 19
RIGID 20
RIGID 21
RIGID 22
RIGID 23
RIGID 24
RIGID 25
RIGID 26
RIGID 27
RIGID 28
RIGID 29
RIGID 30
RIGID 31
RIGID 32
RIGID 33
RIGID 34
RIGID 35
RIGID 36
RIGID 37
RIGID 38
RIGID 39
RIGID 40
RIGID 41
RIGID 42
RIGID 43
RIGID 44
RIGID 45
RIGID 46
RIGID 47
RIGID 48
RIGID 49
RIGID 50
RIGID 51
RIGID 52
RIGID 53

```

40 FORMAT (1H0,20X,*TIME RESPONSE*)
41 FORMAT (1H0,20X,F8.4,*RESPONSE SECS*,2X,F8.4,*HZ COUNTED*)
42 FORMAT (1H0,2X,*TIME *,E(7X),*STA*,13))
43 FORMAT (1H ,F8.4,8(1X,E12.5))
50 FORMAT (1H0,20X,*FREQUENCY RESPONSE*)
52 FORMAT (1H0,19X,*STA*,13,3(20X,*STA*,13)
1 / 5X,*FREQ*,6X,*REAL*,9X,*IMAG*,3(9X,*REAL*,9X,*IMAG*) )

C
      READ (NIN,10) (IBOXES(I),I=1,NBOX)
      READ (NIN,10) ((ISBES(I,J),I=1,2),J=1,NSBE)
      READ (NIN,20) (Q(I),I=1,NSYM)
      WRITE (NOUT,38) (I,Q(I),I=1,NSYM)

C
      FACTOR = SIZECT**2
      RHQA = SIGMA*RHOO
      DYNP = SIGMA*RHOO*VTFPS**2*FACTOR/283.

C
      CALL RDAERO (4HGEOM,0,GEOP,NOUT,NER)
      DO 70 I=1,NBOX
      II = IBOXES(I)
      70 GEOM(II) = GEOP(II)

C
      CALL RDAERO (4HGEOL,0,GEOP,NOUT,NEF)
      DO 75 I=1,NBOX
      GEOP(I,4) = GEOP(I,5) - GEOP(I,2)
      75 GEOP(I,5) = GEOP(I,6) - GEOP(I,3)

C
      CALL RDAERO (4HGEU3,0,GEOP,NOUT,NER)

C
      CALL RDAERO (4HGEOS,0,GEOS,NOUT,NER)
      I = NBOX
      DO 80 N=1,NSBE
      I = I+1
      II = ISBES(1,N)
      80 GFUM(I) = GFUS(II,4) - GFUS(II,1)

C
      CALL HEADNG
      WRITE (NOUT,31)
      DO 82 I=1,NBOX
      N = IBOXES(I)
      X = GEOP(N,1)
      Y = GEOP(N,2)
      Z = GEOP(N,3)
      82 WRITE (NOUT,32) I,IBOXES(I),X,Y,Z,GEOM(I)
      N = NBOX
      WRITE (NOUT,33)
      DO 55 J=1,NSBE
      N = N+1
      I = ISBES(1,J)
      X = (GEOS(I,1)+GEOS(I,4))/2.0
      Y = (GEOS(I,2)+GEOS(I,5))/2.0
      Z = (GEOS(I,3)+GEOS(I,6))/2.0

```

RIGID 54
RIGID 55
RIGID 56
RIGID 57
RIGID 58
RIGID 59
RIGID 60
RIGID 61
RIGID 62
RIGID 63
RIGID 64
RIGID 65
RIGID 66
RIGID 67
RIGID 68
RIGID 69
RIGID 70
RIGID 71
RIGID 72
RIGID 73
RIGID 74
RIGID 75
RIGID 76
RIGID 77
RIGID 78
RIGID 79
RIGID 80
RIGID 81
RIGID 82
RIGID 83
RIGID 84
RIGID 85
RIGID 86
RIGID 87
RIGID 88
RIGID 89
RIGID 90
RIGID 91
RIGID 92
RIGID 93
RIGID 94
RIGID 95
RIGID 96
RIGID 97
RIGID 98
RIGID 99
RIGID100
RIGID101
RIGID102
RIGID103
RIGID104
RIGID105

```

C      55 WRITE (NUUT,34) N, (ISRES(I,J),I=1,2),X,Y,Z,GENDM(N)
C      CALL RDAERO (4HRK ,0,CR,NUUT,NER)
C      GAMX = CR(1,NOR)
C      GAMZ = CR(2,NOR)
C      GAMZ = CR(3,NOR)
C      CALL BLINT (MBOX,MSBE
C      1 , GEOP,GFOS,CAMX,GAMY,GAMZ
C      2 , ALI,X,Y,Z)
C      NOW THE TIME OF BURST INTERCEPT (WRT ORIGIN OF AAS) IS
C      TAU = (ALI*SIZECT/12.)/(VSS*(1.0+GAMX*AMACH))
C      IF (TAU.GT.0) TAU=0.
C      WRITE (NUUT,3) NOR,TIMEMX,NT,RSRSP,GAMX,GAMY,GAMZ,TAU,X,Y,Z
C      CALL RDAERO (4HRK ,0,VCHWS,NUUT,NER)
C      DO 90 I=1,NK
C      IF (VOBWS(I).EQ.0.0) VOBWS(I)=1.E-10
C      VOBWS(I) = 1./VOBWS(I)
C      DO 60 K=1,NFREQ
C      OMEGA(K) = OMEGA(K)*TWGPI
C      IF (OMEGA(K).LE.0.0) OMEGA(K) = 1.E-6
C      60 CONTINUE
C      OMEGMX = (OMEGA(NFREQ)
C      FRQMX = OMFGMX/TWUPI
C      CALL RDAERO (4HDPSP,1,D,NUUT,NER)
C      DO 105 I=1,NBOX
C      II = IBOXES(I)
C      CPT(I) = 0.0
C      DO 105 J=1,NSYM
C      105 CPT(I) = CPT(I) + DYNP*D(II,J)*Q(J)
C      CALL RDAERO (4HDZSP,1,DB,NUUT,NER)
C      DO 120 N=1,NSBE
C      IUIR = ISRES(2,N)
C      IF (IUIR.EQ.2) GO TO 120
C      II = ISRES(1,N)
C      I = N + NBOX
C      CPT(I) = 0.0
C      DO 110 J=1,NSYM
C      110 CPT(I) = CPT(I) + DYNP*DB(II,J)*Q(J)
C      120 CONTINUE
C      CALL RDAERO (4HDYSP,1,DB,NUUT,NER)
C      DO 150 N=1,NSBE

```

```

RIGID106
RIGID107
RIGID108
RIGID109
RIGID110
RIGID111
RIGID112
RIGID113
RIGID114
RIGID115
RIGID116
RIGID117
RIGID118
RIGID119
RIGID120
RIGID121
RIGID122
RIGID123
RIGID124
RIGID125
RIGID126
RIGID127
RIGID128
RIGID129
RIGID130
RIGID131
RIGID132
RIGID133
RIGID134
RIGID135
RIGID136
RIGID137
RIGID138
RIGID139
RIGID140
RIGID141
RIGID142
RIGID143
RIGID144
RIGID145
RIGID146
RIGID147
RIGID148
RIGID149
RIGID150
RIGID151
RIGID152
RIGID153
RIGID154
RIGID155
RIGID156
RIGID157

```

```

IDIR = ISRES(2,N)
IF (IDIR.EQ.1) GO TO 150
II = ISRES(1,N)
I = N + NBOX
CPT(I) = 0.0
DO 140 J=1,NSYM
  CPT(I) = CPT(I) + DYNP*DB(II,J)*Q(J)
140 CONTINUE
C
CALL HEADNG
CALL PRNT (CPT,NFORCE,1,NFORCE,2,12HTRIM FORCES ,3)
C
DO 250 K=1,NK
  CALL RDAERO (4HFPSP,K,D,NCUT,NER)
C
DO 220 I=1,NBOX
  II = IBOXES(I)
  FI(I,K) = D(II,NOR)
220 CONTINUE
C
CALL RDAERO (4HFZSP,K,DB,NCUT,NER)
C
DO 230 N=1,NSBE
  IDIR = ISRES(2,N)
  IF (IDIR.EQ.2) GO TO 230
  I = N+NBOX
  FI(I,K) = DB(II,NOR)
230 CONTINUE
C
CALL RDAERO (4HFYSP,K,DB,NCUT,NER)
C
DO 240 N=1,NSBE
  IDIR = ISRES(2,N)
  IF (IDIR.EQ.1) GO TO 240
  I = N+NBOX
  FI(I,K) = DB(II,NOR)
240 CONTINUE
C
250 CONTINUE
C
CALL GSINT (NFREQ,NFORCE,NK,IPREQ
1 , OMEGA,CP,FI,VOBWS,RS,TANS)
C
LINES = KROW+1
DO 253 JS=1,NFORCE,4
  JF = MINO (JS+3,NFORCE)
  IF (LINES+4.LE.KROW) GO TO 252
  CALL HEADNG
  WRITE (NCUT,50)
  LINES = LINES+2
253 CONTINUE

```

```

252 WRITE (NOUT,52) (J,J=JS,JF)
   LINES = LINES+3
   DO 253 K=1,NFREQ
   FREQ = OMFGA(K)/TWOPI
   LINES = LINES+1
   IF (LINES.LE.KROW) GO TO 253
   CALL HEADNG
   WRITE (NOUT,52) (J,J=JS,JF)
   LINES = LINES+4
253 WRITE (NOUT,43) FREQ, (CP(J,K),J=JS,JF)
      C
      CALL HEADNG
      DO 254 I=1,NTMGST
254 READ (NIN,20) TIME(I), DP(I), RHU(I), VG(I)
   WRITE (NOUT,30) (I,TIME(I),DP(I),RHU(I),VG(I),I=1,NTMGST)
      C
      CALL PLTI ( NTMGST,TIME,VG,I)
      WRITE (NOUT,255)
255 FORMAT (1H0,20X,*INPUT GLST TIME HISTORY*)
      C
      DO 256 I=1,NTMGST
256 TIME(I) = TIME(I) - TAU
      C
      CALL TKFFT (FTG,NTMGST,OMEGA,VG,RHU,RHUA,NFREQ
1 , PLCTF,PLGTA,TIME,GZERG,ALPHA,BETA)
      C
      IF (IPLST.EQ.0) GO TO 290
      C
      DO 260 J=1,NFREQ
      PLOT(J) = OMEGA(J)/TWOPI
260 PLOTA(J) = SORT(REAL(FTG(J))*2+AIMAG(FTG(J))*2)
      C
      CALL PLTI(NFREQ,PLOTF,PLOTA,I)
      WRITE (NOUT,261)
261 FORMAT (1H0,20X,21HTRANSFORM OF RHO*GUST )
      C
      DO 300 J=1,NFREQ
      DO 300 I=1,NFORCE
300 CP(I,J) = CP(I,J)*FTG(J)*DYNP/VTFPS
      C
      NOW GET SOLN TIMES
      IND=0
      DELTT=DELT
      TI=-DELT
      DO 100 I=1,NTMRSP
      TI=TI+DELT
      TIME(I)=TI
      IF (IND.GT.1) GO TO 100
      IF (TI.LT.0.25) GO TO 100
      IF (TI.LE.1.0) GO TO 99
      IND=2
      TI=TI-DELT

```

RIGID210
 RIGID211
 RIGID212
 RIGID213
 RIGID214
 RIGID215
 RIGID216
 RIGID217
 RIGID218
 RIGID219
 RIGID220
 RIGID221
 RIGID222
 RIGID223
 RIGID224
 RIGID225
 RIGID226
 RIGID227
 RIGID228
 RIGID229
 RIGID230
 RIGID231
 RIGID232
 RIGID233
 RIGID234
 RIGID235
 RIGID236
 RIGID237
 RIGID238
 RIGID239
 RIGID240
 RIGID241
 RIGID242
 RIGID243
 RIGID244
 RIGID245
 RIGID246
 RIGID247
 RIGID248
 RIGID249
 RIGID250
 RIGID251
 RIGID252
 RIGID253
 RIGID254
 RIGID255
 RIGID256
 RIGID257
 RIGID258
 RIGID259
 RIGID260
 RIGID261

```

DELT=DELT*2.0
TI=TI+DELT
TIME(I)=TI
GO TO 100
IF (IND.EQ.1) GO TO 100
IND=1
TI=TI-DELT
DELT=DELT*5.0
TI=TI+DELT
TIME(I)=TI
100 CONTINUE
C
IF (TIME(NIMRSP).GT.TIMEMX) TIME(NIMRSP)=TIMEMX
C
CALL IFT (NFREQ,OMEGA,CMFGMX,0.0,0,NFORCE,CP,NTMRSP,TIME,P,TIMEMX)
C
DO 310 I=1,NFORCE
DO 310 J=1,NTMRSP
310 P(I,J) = (P(I,J) + CPT(I))/GFQM(I)
C
LINES = KROW+1
DO 820 JS=1,NFORCE,8
JF = MIN(JS+7,NFORCE)
IF (LINES+3.LE.KROW) GO TO 810
CALL HEADNG
WRITE (NOUT,40)
WRITE (NOUT,41) TIMEMX,FKQMX
LINES = LINES+4
810 WRITE (NOUT,42) (J,J=JS,JF)
LINES = LINES+2
DO 820 K=1,NTMRSP
LINES = LINES+1
IF (LINES.LE.KROW) GO TO 820
CALL HEADNG
WRITE (NOUT,42) (J,J=JS,JF)
LINES = LINES+3
820 WRITE (NOUT,43) TIME(K), (P(J,K),J=JS,JF)
C
IF (IPLT.EQ.0) GO TO 900
DO 850 I=1,NFORCE
DO 840 J=1,NTMRSP
840 PLOTA(J) = P(I,J)
CALL PLOT1 (NTMRSP,TIME,PLOTA,1)
WRITE (NOUT,830) I
830 FORMAT (1H0,20X,*TIME RESPONSE FOR STATION*,I4)
850 CONTINUE
C
900 RETURN
END

```

```

C      SURROUTINE CMERGE
C      COMMON NAA,A(1)
C      COMMON /ZZZ/HEDR(48),NIN,NOUT,KROW,LINFS,IPRNT,NER
C
C      DIMENSION IHD(50)
C
C      10 FORMAT (6I12)
C      20 FORMAT (6F12.0)
C      30 FORMAT (///10X,*,LIST OF REDUCED FREQUENCIES TO BE MERGED FROM TAPE
C      1E FT#,I2/(5X,6F15.6))
C
C      NTAPE1 = 17
C      NTAPE2 = 18
C      NTAPE3 = 19
C      REWIND NTAPE1
C      REWIND NTAPE2
C      REWIND NTAPE3
C
C      READ (NIN,10) INK1, INK2
C
C      READ (NTAPE1) IHD
C      NK1 = IHD(2)
C      READ (NTAPE2) IHD
C      NK2 = IHD(2)
C
C      L1 = NAA
C      L2 = L1 + INK1
C      L3 = L2 + INK2
C      L4 = L3 + INK1 + INK2
C      L5 = L4 + NK1
C      L6 = L5 + NK2
C
C      CALL HEADNG
C      LAST = L2-1
C      READ (NIN,20)
C      WRITE (NOUT,30) NTAPE1,(A(I),I=L1,LAST)
C      LAST = L3-1
C      READ (NIN,20)
C      WRITE (NOUT,30) NTAPE2,(A(I),I=L2,LAST)
C
C      CALL MERGE (NTAPE1,NTAPE2,NTAPE3,NOUT,INK1,INK2
C      1 , A(L1),A(L2),A(L3),A(L4),A(L5),A(L6) )
C
C      RETURN
C      END

```

2 3 4 5 6 7 8 9

COPY
COPY
COPY
COPY
COPY
COPY
COPY
COPY

SUBROUTINE COPY(A , L , NI , NO)
DIMENSION A (L)
REAC (NI) A
WRITE (NO) A
RETURN
END

C

C

FCOPY 54
FCOPY 55
FCOPY 56
FCOPY 57
FCOPY 58
FCOPY 59
FCOPY 60
FCOPY 61
FCOPY 62
FCOPY 63
FCOPY 64
FCOPY 65
FCOPY 66
FCOPY 67
FCOPY 68
FCOPY 69
FCOPY 70
FCOPY 71
FCOPY 72
FCOPY 73
FCOPY 74
FCOPY 75
FCOPY 76
FCOPY 77
FCOPY 78
FCOPY 79
FCOPY 80
FCOPY 81
FCOPY 82
FCOPY 83
FCOPY 84
FCOPY 85
FCOPY 86
FCOPY 87
FCOPY 88
FCOPY 89
FCOPY 90
FCOPY 91
FCOPY 92
FCOPY 93
FCOPY 94
FCOPY 95
FCOPY 96
FCOPY 97
FCOPY 98
FCOPY 99
FCOPY100
FCOPY101
FCOPY102
FCOPY103
FCOPY104
FCOPY105

```

80 CONTINUE
C      SKIP OVER A SET OF MATRICES THAT IS NOT TO BE SAVED
C
C      DO 90 ISK = 1, ISKIP
C      READ (NTAPE)
C      90 CONTINUE
C
C      GO TO (40,50), IGO
C      100 CONTINUE = 2
C      IGO
C      NTAPE = NTAPE2
C      KKRUN = KK2(K2)
C      GO TO 50
C
C      110 CONTINUE
C      COPY THIS SET OF FORCE- AND GENERALIZED FORCE MATRICES
C
C      NREC = NSYM + NGUST
C      DO 140 I = 1, 2
C      LENGTH = NBOX * 2
C      DO 130 NST = 1, NSET
C      DO 120 NRC = 1, NREC
C
C      CALL COPY( TEMP, LENGTH, NTAPE, NTAPE3 )
C
C      120 CONTINUE
C      IF (NSBE .NF. 0) LENGTH = NS3E * 2
C      130 CONTINUE
C
C      IF (NASYM .EQ. 0) GO TO 150
C      NREC = NASYM + NGUST
C      140 CONTINUE
C
C      150 CONTINUE = NSYM * (NSYM + NGUST) * 2
C      NDIM = 1
C      DO 170 I = 1, 2
C      DO 160 N = 1, NSET
C
C      CALL COPY( TEMP, NDIM, NTAPE, NTAPE3 )
C
C      160 CONTINUE
C      IF (NASYM .EQ. 0) GO TO 180
C      NDIM = NASYM * (NASYM + NGUST) * 2
C      170 CONTINUE
C
C      180 CONTINUE
C
C      IF (JK .EQ. NK) GO TO 210
C      IF (INCR .EQ. 1) GO TO 220
C      GO TO (190,200), IGO

```

FCOPY106
FCOPY107
FCOPY108
FCOPY109
FCOPY110
FCOPY111
FCOPY112
FCOPY113
FCOPY114
FCOPY115
FCOPY116
FCOPY117
FCOPY118
FCOPY119
FCOPY120
FCOPY121
FCOPY122
FCOPY123
FCOPY124
FCOPY125
FCOPY126
FCOPY127
FCOPY128
FCOPY129
FCOPY130
FCOPY131
FCOPY132
FCOPY133
FCOPY134
FCOPY135
FCOPY136
FCOPY137
FCOPY138
FCOPY139
FCOPY140
FCOPY141

```

190 IF (K1.EQ.NK1) GO TO 220
    K1 = K1 + 1
    GO TO 210
C
200 IF (K2.EQ.NK2) GO TO 220
    K2 = K2 + 1
205 FORMAT (1H,*FORCES WRITTEN TO OUTPUT FOR K=*,F10.6)
210 WRITE (NTP6,205) FREQ
C
    END OF FREQUENCY LOOP
C
    RETURN
C
220 CONTINUE
C
    REWIND NTAPE AND POSITION IT FOR THE READING OF THE FIRST
    SFT OF FORCE- AND GENERALIZED FORCE MATRICES
C
    REWIND NTAPE
    DO 230 NSK = 1, NSKIP
    REAC (NTAPE)
    230 CONTINUE
C
    GO TO (240,250), IGO
    240 CONTINUE = IGO1 + 1
    K1 = 1
    IF (FREQ.EQ.RKKUN) GO TO 210
    GO TO 40
    250 CONTINUE = IGO2 + 1
    K2 = 1
    IF (FREQ.EQ.RKKUN) GO TO 210
    GO TO 100
C
    END

```

```

C      SUBROUTINE MERGE(NTAPE1, NTAPE2, NTAPE3, NTP6, INK1, INK2, TEMP)
C      RKINI, PKIN2, RK, RK1, RK2,
C      DIMENSION IDH1(50), IDH2(50), RKINI(1), RKIN2(1), NDIM(10)
C      RK1(1), RK2(1), RK(1), TEMP(1)
C      ( //10X, 32H*** INCOMPATIBLE INPUT TAPES *** / )
C      10 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      20 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      30 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      40 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      50 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      60 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      70 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      80 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      90 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      100 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      110 FORMAT ( //10X, 9HNSYM1 =, 14, 11H NSYM2 =, 14, 7H -STOP- )
C      12 // 10X, 20HCASE IS TERMINATED
C      NK = INK1 + INK2
C      REWIND NTAPE1
C      REWIND NTAPE2
C      REWIND NTAPE3
C      READ (NTAPE1) IDH1
C      READ (NTAPE2) IDH2
C      DO 120 I = 3, 8
C      IF (IDH1(I) .EQ. IDH2(I)) GO TO 120
C      IX = I - 2
C      GO TO 130
C      120 CONTINUE
C      GO TO 200
C      130 CONTINUE
C      WRITE (NTP6, 10)
C      GO TO (140, 150, 160, 170, 180, 190), IX
C      140 WRITE (NTP6, 20) IDH1(3), IDH2(3)
C      STOP
C      150 WRITE (NTP6, 30) IDH1(4), IDH2(4)
C      STOP
C      160 WRITE (NTP6, 40) IDH1(5), IDH2(5)
C      STOP
C      170 WRITE (NTP6, 50) IDH1(6), IDH2(6)
C      STOP
C      180 WRITE (NTP6, 60) IDH1(7), IDH2(7)
C      STOP
C      190 WRITE (NTP6, 70) IDH1(8), IDH2(8)
C      STOP

```

```

200 CONTINUE
C      NK1 = IDH1(2)
      NK2 = IDH2(2)
      IDH1(2) = NK
      CALL WK AERO (4HHEAD,0, IDH1, NTP6, NEK)

C      NSYM = IDH1(3)
      NASYM = IDH1(4)
      NGUST = IDH1(5)
      NB = IDH1(6)
      NBOX = IDH1(7)
      NSBE = IDH1(8)
      NSTRIP = IDH1(9)
      MAXSTR = IDH1(10)
      NP = IDH1(11)

C      NGFOM = 3*NBOX + 8*NSTRIP + MAXSTR + 2*NP + 3*NB
      NDEM = NCEOM
      NDM(1) = 6 * NBOX
      NDM(2) = 3 * NBOX
      NDM(3) = 6 * NSBE
      NDM(4) = 3 * NGUST
      NDM(5) = 3 * NGUST

C      COPY RECORD TYPES 2 THROUGH 6 FROM NTAPE1 ONTO NTAPE3
C
C      DO 210 J = 1, 5
      LENGTH = NDM(J)
      IF (LENGTH.EQ. 0) GO TO 210
      READ (NTAPE2)
      CALL COPY( TEMP, LENGTH, NTAPE1, NTAPE3 )

C      210 CONTINUE
C      READ FREQUENCY ARRAYS FROM THE TWO INPUT TAPES
      READ (NTAPE1) (RK1(I), I = 1, NK1)
      READ (NTAPE2) (RK2(I), I = 1, NK2)

C      CHECK CARD-INPUT FREQUENCY ARRAYS AGAINST FREQUENCIES THAT
      ARE AVAILABLE ON THE TWO INPUT TAPES
      NTAPE = NTAPE1
      INK = INK1
      NKR = NK1
      DO 310 IGO = 1, 2
      NKMAX = 0
      DO 300 K = 1, INK
      GO TO (20,230), IGO
      FREQ = RKIN1(K)
      220 GO TO 240

```

```

MERGE 54
MERGE 55
MERGE 56
MERGE 57
MERGE 58
MERGE 59
MERGE 60
MERGE 61
MERGE 62
MERGE 63
MERGE 64
MERGE 65
MERGE 66
MERGE 67
MERGE 68
MERGE 69
MERGE 70
MERGE 71
MERGE 72
MERGE 73
MERGE 74
MERGE 75
MERGE 76
MERGE 77
MERGE 78
MERGE 79
MERGE 80
MERGE 81
MERGE 82
MERGE 83
MERGE 84
MERGE 85
MERGE 86
MERGE 87
MERGE 88
MERGE 89
MERGE 90
MERGE 91
MERGE 92
MERGE 93
MERGE 94
MERGE 95
MERGE 96
MERGE 97
MERGE 98
MERGE 99
MERGE 100
MERGE 101
MERGE 102
MERGE 103
MERGE 104
MERGE 105

```

MERGE106
MERGE107
MERGE108
MERGE109
MERGE110
MERGE111
MERGE112
MERGE113
MERGE114
MERGE115
MERGE116
MERGE117
MERGE118
MERGE119
MERGE120
MERGE121
MERGE122
MERGE123
MERGE124
MERGE125
MERGE126
MERGE127
MERGE128
MERGE129
MERGE130
MERGE131
MERGE132
MERGE133
MERGE134
MERGE135
MERGE136
MERGE137
MERGE138
MERGE139
MERGE140
MERGE141
MERGE142
MERGE143
MERGE144
MERGE145
MERGE146
MERGE147
MERGE148
MERGE149
MERGE150
MERGE151
MERGE152
MERGE153
MERGE154
MERGE155
MERGE156
MERGE157

```

230 FREQ = RKIN2(K)
C
240 CONTINUE
C
DO 280 J = 1, NKR
  JMAX = J
  GO TO (250,260), IGO
250 RPICK = RK1(J)
  GO TO 270
260 RPICK = RK2(J)
C
270 CONTINUE
  IF (FREQ.EQ.RPICK) GO TO 290
C
280 CONTINUE
C
      THIS (CARD-INPUT) FREQUENCY IS NOT AVAILABLE ON TAPE -- STOP
C
      WRITE (NTP6,110) FREQ, NTAPE
C
      STOP
C
290 IF (JMAX.GT.NKMAX) NKMAX = JMAX
300 CONTINUE
  IF (IGD.EQ.1) NK1 = NKMAX
  IF (IGU.EQ.2) NK2 = NKMAX
  NTAPE = NTAPE2
  INK = INK2
  NKR = NK2
C
310 CONTINUE
C
      MOVE THE TWO INPUT-FREQUENCY-ARRAYS INTO TEMP, THEN SORT
      IN ASCENDING ORDER AND SAVE IN ARRAY RK
C
DO 330 K = 1, NK
  IF (K.GT. INK1) GO TO 320
  TEMP(K) = RKIN1(K)
  GO TO 330
C
320 CONTINUE = K - INK1
  TEMP(K) = RKIN2(1)
330 CONTINUE = NK
  J
340 CONTINUE = 0.0
  RKMAX = 0.0
  DO 350 K = 1, NK
    IF (RKMAX.LT. TEMP(K)) RKMAX = TEMP(K)
350 CONTINUE
C
  RK(J) = RKMAX

```

```

IF (J.EQ.1) GO TO 360
DO 352 K = 1, NK
  L = K
  IF (RKMAX.EQ. TEMP(L)) GO TO 354
  352 CONTINUE
  STOP 16
C
  354 TEMP(L) = 0.0
C
  J = J - 1
  GO TO 340
C
  360 CONTINUE
C
  WRITE (NTAPE3) (RK(I), I = 1, NK)
C
  COPY THE MATRICES HPS, HPA, HZS, HZA, HYS, HYA
C
  LENGTH = NBOX
  DO 400 K = 1, 3
    NREC = NSYM
    DO 380 I = 1, 2
      DO 370 N = 1, NREC
        READ (NTAPE2)
        CALL COPY( TEMP, LENGTH, NTAPE1, NTAPE3 )
C
        370 CONTINUE
C
        IF (I.EQ.2) GO TO 380
        IF (NASYM.EQ.0) GO TO 390
        NREC = NASYM
        CONTINUE
        380 CONTINUE
        390 CONTINUE
        IF (NSBE.EQ.0) GO TO 410
        LENGTH = NSBE
        CONTINUE
        400 CONTINUE
C
        410 CONTINUE
        WRITE (NTP6,420) NTAPE3, (RK(I), I = 1, NK)
        420 FORMAT ( //10X,45HHEADER-ITEMS, GEOMETRY, FREQUENCY ARRAY AND ,
          1 36HTHE H-MATRICES ARE SAVED ON TAPE FT, 12 ///10X
          2 55HBEGIN MERGING THE FORCE- AND GENERALIZED FORCE MATRICES
          3 //10X, 47HTHE COMBINED (SORTED) OUTPUT-FREQUENCY-ARRAY IS
          4 // (10X, 6F10.6) )
C
        COPY FORCES AND GENERALIZED FORCES --
C
        CALL FCOPY (NTAPE1, NTAPE2, NTAPE3, NK1
          1 RK1 , RK2 , RK , NSYM , NASYM , NK2 , NGUST , NK , NBOX ;

```

```

MERGE158
MERGE159
MERGE160
MERGE161
MERGE162
MERGE163
MERGE164
MERGE165
MERGE166
MERGE167
MERGE168
MERGE169
MERGE170
MERGE171
MERGE172
MERGE173
MERGE174
MERGE175
MERGE176
MERGE177
MERGE178
MERGE179
MERGE180
MERGE181
MERGE182
MERGE183
MERGE184
MERGE185
MERGE186
MERGE187
MERGE188
MERGE189
MERGE190
MERGE191
MERGE192
MERGE193
MERGE194
MERGE195
MERGE196
MERGE197
MERGE198
MERGE199
MERGE200
MERGE201
MERGE202
MERGE203
MERGE204
MERGE205
MERGE206
MERGE207
MERGE208
MERGE209

```

MERGE210
MERGE211
MERGE212
MERGE213
MERGE214

1

TEMP

NSBE , NTP6 ,

2 RETURN
END

C C

